

AD-A056 212

NATIONAL MILITARY COMMAND SYSTEM SUPPORT CENTER WASH--ETC F/G 9/2  
NMCS INFORMATION PROCESSING SYSTEM 360 FORMATTED FILE SYSTEM (N--ETC(U)  
JUN 75

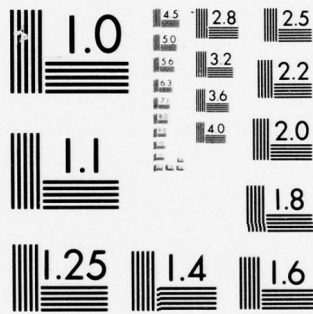
UNCLASSIFIED

NMCSSC-CSM-15-74-V8-1/2/3

NL

1 OF 2  
AD  
A056212





MICROCOPY RESOLUTION TEST CHART  
NATIONAL BUREAU OF STANDARDS-1963-A





DEFENSE COMMUNICATIONS AGENCY  
NATIONAL MILITARY COMMAND SYSTEM  
SUPPORT CENTER

WASHINGTON, D. C. 20301

NMCS Information Processing System  
360 Formatted File System (NIPS 360 FFS)  
- Users Manual. Volume VIII. Job  
Preparation. Changes 1, 2, and 3.

1 June 1975

LEVEL

IN REPLY  
REFER TO:

TO: DISTRIBUTION

SUBJECT: Change 1 to CSM UM 15-74, Job Preparation  
Volume VIII, dated 15 October 1974

14/NMCS SC-CSM-15-74-V8-1/2/3

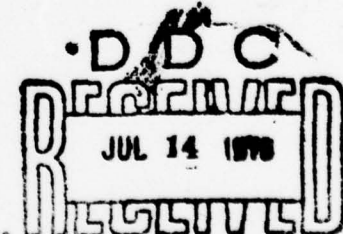
1. Insert the enclosed change pages and destroy the replaced pages according to applicable security regulations.
2. A list of Effective Pages to verify the accuracy of this manual is enclosed. This list should be inserted before the title page.
3. When this change has been posted, make an entry in the Record of Changes on the inside cover.

FOR THE COMMANDER:

45 Enclosures  
Change 1 pages

J. DOUGLAS POTTER  
Chief, Military Personnel  
and Administrative  
Services Office

THIS DOCUMENT IS BEST QUALITY PRACTICABLE.  
THE COPY FURNISHED TO DDC CONTAINED A  
SIGNIFICANT NUMBER OF PAGES WHICH DO NOT  
REPRODUCE LEGIBLY.



DISTRIBUTION STATEMENT A

Approved for public release  
Distribution Unlimited

AD A056212

DDC FILE COPY

243 000

JCB

## **DISCLAIMER NOTICE**

**THIS DOCUMENT IS BEST QUALITY  
PRACTICABLE. THE COPY FURNISHED  
TO DDC CONTAINED A SIGNIFICANT  
NUMBER OF PAGES WHICH DO NOT  
REPRODUCE LEGIBLY.**

# EFFECTIVE PAGES - 1 June 1975

This list is used to verify the accuracy of CSM UM 15-74, Volume VIII, after change pages have been inserted. Original pages are indicated by the letter O, change 1 by the numeral 1.

<u>Page No.</u>	<u>Change No.</u>
TITLE	O
ii - iv	O
v	1
vi - 106	O
107 - 136	1
137 - 142	O
143 - 156	1
157 - 162	O

APPROVED BY		
OTIS	State Control	<input checked="" type="checkbox"/>
DD	Gov Control	<input type="checkbox"/>
CLASSIFIED		<input type="checkbox"/>
AUTOMATIC		
BY		
DISTRIBUTION/AVAILABILITY CODES		
Dist.	AVAIL. ENG. OF SPECIAL	
A	23 EJL	

**Section**

**Page**

**APPENDIX**

<b>A</b>	<b>SYMBOLIC PARAMETER DEFINITIONS.....</b>	<b>62</b>
<b>B</b>	<b>INPUT SOURCE DDNAME.....</b>	<b>71</b>
<b>C</b>	<b>PROCEDURES DESCRIPTIONS.....</b>	<b>71</b>
<b>D</b>	<b>NIPS PROCEDURE RETURN CODES.....</b>	<b>74</b>
<b>E</b>	<b>NIPS PROCEDURE DD STATEMENT USAGE.....</b>	<b>76</b>
<b>#F</b>	<b>PROCEDURE LISTINGS.....</b>	<b>107</b>

XCLASS  
 XDMPLIB  
 XPM  
 XPMEX  
 XFR  
 XFS  
 XISTOS  
 XKA  
 XKM  
 XOP  
 XOP EX  
 XOP SD  
 XOPSD EX  
 XQRTQ DF  
 XQUIP  
 XQUIP SD  
 XRASP  
 XRA SPEX  
 XRESTANS  
 XRESTLIB  
 XSAVEANS  
 XSAVELIB  
 XSP  
 XSTOIS  
 XSUBCHK  
 XSUBLDR  
 XTABGEN  
 XTP  
 XTIPMON  
 XTP SUP

## JOB PREPARATION

### Appendix F PROCEDURE LISTINGS



## NIPS 360 FFS

## PROCEDURES

//XCLASS	PROC A=A,BSZFILE=,BSZNEWF=,	00000100
//	CL=',' ,CL1=',' ,DEN=,ISAM='DUMMY.FILE',	00000200
//	JOB LIB='FFS.JOB LIB',LAB=SL,NDISP=KEEP, RGN=60K,	00000300
//	PTFJOB L='PTF.JOB LIB',	00000400
//	SAM='DUMMY.FILE',SAMOUT='DUMMY',' ,TRCH=,	00000500
//	UISAM=2314,USAM='(TAPE9,,DEFER)',	00000600
//	VISAM=,VSAM=,VSMOUT=	00000700
/**		00000800
/**	CHARLES W. HICKISCH MAJOR,USA PR CJ CODE=763NIPS BRANCH=431	00000900
/**	DATE=MARCH 1,1974	00001000
/**		00001100
//CLASS	EXEC PGM=UTCLASS,REGION=&RGN	00001200
//STEPLIB DD	DSN=&PTFJOB L,DISP=SHR	00001300
//	DD DSN=&JOB LIB,DISP=SHR	00001400
//SYSPRINT DD	SYSOUT=( &A,&CL )	00001500
//SYSUDUMP DD	SYSOUT=( &A,&CL1 )	00001600
//DATAFILE DD	DSN=&ISAM,UNIT=&UISAM,VOL=&VISAM,DISP=SHR	00001700
//SAMFILE DD	DSN=&SAM.S,UNIT=&USAM,VOL=&VSAM,DISP=( SHR,KEEP),	00001800
//	DCB=(RECFM=VB,LRECL=1000,BLKSIZE=&BSZFILE,TRTCH=&TRCH,	00001900
//	DEN=&DEN),LABEL=( ,&LAB)	00002000
//UTSAMOUT DD	&SAMOUT.DCB=(RECFM=VB,LRECL=1000,BLKSIZE=&BSZNEWF,	00002100
//	TRTCH=&TRCH,DEN=&DEN),DISP=( ,&NDISP),LABEL=( ,&LAB),	00002200
//	UNIT=&USAM,VOL=&VSMOUT	00002300

## NIPS 360 FFS

## PROCEDURES

```

//XDMPLIB      PROC  A=A,BSZFILE=,                00000100
//              CL=',' ,CL1=',' ,DEN=,ISAM='DUMMY.FILE', 00000200
//              JOBLIB='FFS.JOBLIB',LAB=SL,RGN=60K,      00000300
//              PTFJOBL='PTF.JOBLIB',                  00000400
//              SAM='DUMMY.FILE',STG=NIPW,TRCH=,          00000500
//              UISAM='(2314,P)',USAM='(TAPE9,,DEFER)',    00000600
//              VISAM=,VSAM=                             00000700
//**                                                    00000800
//** CHARLES W. HICKISCH  MAJOR,USA  PROJ CODE=763NIPS  BRANCH 431 00000900
//** DATE=MARCH 1,1974                                00001000
//**                                                    00001100
//UTDMP      EXEC  PGM=UTDMPLIB,REGICN=&RGN             00001200
//STEPLIB DD DSN=&PTFJOBL,DISP=SHR                     00001300
//          DD DSN=&JOBLIB,DISP=SHR                     00001400
//SYSPRINT DD SYSOUT=(&A,&CL)                          00001500
//SYSUDUMP DD SYSOUT=(&A,&CL1)                          00001600
//DATAFILE DD  DSNAME=&ISAM,UNIT=&UISAM,VOLUME=&VISAM,DISP=SHR 00001700
//ISAMWORK DD  UNIT=&STG,DCB=DSORG=IS,SPACE=(CYL,(10))    00001800
//SAMFILE DD  DSNAME=&SAM.S,UNIT=&USAM,VOLUME=&VSAM,DISP=(SHR,KEEP), C00001900
//          LABEL=(,&LAB),CCB=(RECFM=VB,LRECL=1000,BLKSIZE=&BSZFILE, 00002000
//          TRCH=&TRCH,DEN=&DEN)                          00002100

```

## NIPS 360 FFS

## PROCEDURES

```

//XFM  PROC  A=A,AUXBUFN=2,AUXSP=6,BLKSIZE=560,B=B,          00000100
//          BSZFILE=,BSZNEW=,                                X00000200
//          CL=',' ,CL1=',' ,CL2=',' ,CYLOFL=1,DEN=,          00000300
//          EROPT=ABE,GEN='DUMMY',,                          00000400
//          INDEX=1,ISAM='DUMMY.FILE',                        00000500
//          INXSP=1,JOBLIB='FFS.JOBLIB',JOBMAC='FFS.JOBMACRO', 00000600
//          LAB=SL,LIB='DUMMY.FILE',LIB1='DUMMY.FILE',        X00000700
//          MCTSP=5,NBRBLK=200,QVFLOW=1,PRIME=5,RGN=98K,      00000800
//          PTFJOBL='PTF.JOBLIB',PTFJOBM='PTF.JOBMACRO',      00000900
//          SAM='DUMMY.FILE',SAMOUT='DUMMY',,SDISP=SHR,        00001000
//          SOURCL='DUMMY.FILE',STG=NIPW,TDISP=MCD,TRANS='&&TRANS', 00001100
//          TRANSP=200,TRBUFNO=4,TRCH=,UISAM='(2314,P)',      00001200
//          ULIB=2314,ULIB1=2314,USAM='(TAPE9,,DEFER)',        00001300
//          USOURCL=2314,UTRANS=NIPW,VINDEX='REF=*.DATAFILE',  00001400
//          VISAM=,VLIB=,VLIB1=,VCVFLOW='REF=*.DATAFILE',     00001500
//          VPRIME='REF=*.DATAFILE',VSAM=,VSMCT=,VSMOUT=,VSOURCL=, 00001600
//          VTRANS=,XDISP=SHR,XINDEX='DUMMY.FILE',            00001700
//          XUNIT=2314,XVOL=                                   00001800
//**                                                    00001900
//** CHARLES W. HICKISCH MAJOR,USA PROJ CODE=763NIPS BRANCH=431 00002000
//** DATE=MARCH 1, 1974                                00002100
//**                                                    00002200
//FM      EXEC  PGM=FM,REGION=&RGN                          00002300
//STEPLIB DD DSN=&PTFJOBL,DISP=SHR                          00002400
//          DD DSN=&JOBLIB,DISP=SHR                          00002500
//FMSTAT  DD SYSOUT=(&A,&ECL)                                00002600
//SYSOUT  DD SYSOUT=(&A,&ECL)                                00002700
//SYSLIST DD SYSOUT=(&A,&ECL)                                00002800
//SOURCPT DD SYSOUT=(&A,&ECL)                                00002900
//SYSUDUMP DD SYSOUT=(&A,&ECL1)                              00003000
//SYSPRINT DD SYSOUT=(&A,&ECL2)                              00003100
//DATAFILE DD DSNNAME=&ISAM,UNIT=&UISAM,VOLUME=&VISAM,DISP=SHR 00003200
//NEWFILE DD &GEN.DCB=(DSORG=IS,CYLOFL=&CYLOFL,BLKSIZE=&BSZNEW), X00003300
//          SPACE=(CYL,&INDEX),                                X00003400
//          DSNNAME=&ISAM.A(INDEX),UNIT=&UISAM,VOLUME=&VINDEX  X00003500
//          DD &GEN.DCB=*.NEWFILE,SPACE=(CYL,&PRIME),        X00003600
//          DSNNAME=&ISAM.A(PRIME),UNIT=&UISAM,VOLUME=&VPRIME, X00003700
//          DISP=(,KEEP,DELETE)                                00003800
//          DD &GEN.DCB=*.NEWFILE,SPACE=(CYL,&CVFLOW),        X00003900
//          DSNNAME=&ISAM.A(CVFLOW),UNIT=&UISAM,VOLUME=&VOVFLOW, X00004000
//          DISP=(,KEEP,DELETE)                                00004100
//FMCOMM  DD UNIT=&STG,SPACE=(TRK,(1,1))                      00004200
//FMFLUD  DD UNIT=&STG,SPACE=(CYL,(&MCTSP,5))                 00004300
//FMTRANS DD UNIT=&STG,SPACE=(CYL,(&AUXSP,5)),DCB=BUFNO=&TRBUFNO 00004400
//FMLABELS DD UNIT=&STG,SPACE=(CYL,(1,1))                     00004500
//FMCM  DD UNIT=&STG,SPACE=(TRK,(20,5))                       00004600
//FMCMFILE DD UNIT=&STG,SPACE=(TRK,(20,5))                    00004700
//FMLITFIL DD UNIT=&STG,SPACE=(TRK,(20,5))                    00004800
//SYSLIB DD DSN=&PTFJOBM,DISP=SHR                              00004900
//          DD DSN=&JOBMAC,DISP=SHR                            00005000
//          DD DSNNAME=SYS1.MACLIB,DISP=SHR                   00005100
//FMLEIN  DD UNIT=&STG,SPACE=(TRK,(25,5))                     00005200
//SYSLMOD DD UNIT=&STG,SPACE=(CYL,(4,1,10)),DCB=SYS1.LINKLIB, X00005300
//          LABEL=EXPDT=66366                                  00005400
//TEMLSREC DD UNIT=&STG,SPACE=(TRK,(20,5))                    00005500
//SORTLIB DD DSNNAME=SYS1.SORTLIB,DISP=SHR                    00005600
//SORTIN  DD UNIT=&STG,SPACE=(TRK,(&TRANSP,150)),DCB=BUFNO=&TRBUFNO 00005700

```



```

//SORTWK01 DD SPACE=(CYL,(&SORTSP),,CONTIG),UNIT=(&STG,SEP=SCRTWK01) 00005800
//SORTWK02 DD SPACE=(CYL,(&SORTSP),,CONTIG), X00005900
// UNIT=(&STG,SEP=SCRTWK01),SEP=SCRTWK01 00006000
//SORTWK03 DD SPACE=(CYL,(&SORTSP),,CONTIG), X00006100
// UNIT=(&STG,SEP=SCRTWK02),SEP=SCRTWK02 00006200
//SORTWK04 DD SPACE=(CYL,(&SORTSP),,CONTIG), X00006300
// UNIT=(&STG,SEP=(SCRTWK01,SCRTWK03)), X00006400
// SEP=(SCRTWK01,SCRTWK03) 00006500
//SORTWK05 DD SPACE=(CYL,(&SORTSP),,CONTIG), X00006600
// UNIT=(&STG,SEP=(SCRTWK02,SCRTWK04)), X00006700
// SEP=(SCRTWK02,SCRTWK04) 00006800
//SORTWK06 DD SPACE=(CYL,(&SORTSP),,CONTIG), X00006900
// UNIT=(&STG,SEP=(SCRTWK01,SCRTWK03,SCRTWK05)), X00007000
// SEP=(SCRTWK01,SCRTWK03,SCRTWK05) 00007100
//SORTOUT DD DISP=(OLD,PASS),DSNAME=*.SORTIN,VOLUME=REF=*.SORTIN, *00007200
// DCB=(LRECL=1500,BLKSIZE=1504,RECFM=VB,BUFNO=2) 00007300
//FMLEFILE DD DSNAME=*.SORTWK04,VOLUME=REF=*.SORTWK04,DISP=(OLD,PASS) 00007400
//FMMACRO DD DSNAME=*.SORTWK05,VOLUME=REF=*.SORTWK05,DISP=(OLD,PASS) 00007500
//FMOUTMAC DD DSNAME=*.SORTWK06,VOLUME=REF=*.SORTWK06,DISP=(OLD,PASS) 00007600
//TAPEIN DD DSNAME=*.SORTIN,VOLUME=REF=*.SORTIN,DISP=(OLD,PASS) 00007700
//TAPEOUT DD DSNAME=*.SORTIN,VOLUME=REF=*.SORTIN,DISP=(OLD,PASS), *00007800
// DCB=(LRECL=1500,BLKSIZE=1504,RECFM=VB) 00007900
//RECSIN DD DSNAME=*.FMFLUD,VOLUME=REF=*.FMFLUD,DISP=(OLD,PASS) 00008000
//RECSOUT DD DSNAME=*.RECSIN,VOLUME=REF=*.RECSIN,DISP=(OLD,PASS), X00008100
// DCB=(LRECL=1000,BLKSIZE=1004,RECFM=VB) 00008200
//RECSWK01 DD DSNAME=*.SORTWK01,VOLUME=REF=*.SORTWK01,DISP=(OLD,PASS) 00008300
//RECSWK02 DD DSNAME=*.SORTWK02,VOLUME=REF=*.SORTWK02,DISP=(OLD,PASS) 00008400
//RECSWK03 DD DSNAME=*.SORTWK03,VOLUME=REF=*.SORTWK03,DISP=(OLD,PASS) 00008500
//RECSWK04 DD DSNAME=*.SORTWK04,VOLUME=REF=*.SORTWK04,DISP=(OLD,PASS) 00008600
//RECSWK05 DD DSNAME=*.SORTWK05,VOLUME=REF=*.SORTWK05,DISP=(OLD,PASS) 00008700
//RECSWK06 DD DSNAME=*.SORTWK06,VOLUME=REF=*.SORTWK06,DISP=(OLD,PASS) 00008800
//TRANS DD DUMMY,DISP=OLD,DCB=(EROPT=&EROPT,BUFNO=&TRBUFN) 00008900
//FMSETTBL DD UNIT=&STG,SPACE=(TRK,(1,1)) 00009000
//FMAUXOP DD DSN=*.FMTRANS,VOLUME=REF=*.FMTRANS,DISP=(OLD,PASS), 00009100
// DCB=(LRECL=1000,BLKSIZE=1004,BUFNO=&AUXBUFN) 00009200
//PUNCH DD SYSOUT=&B,DCB=BUFNO=&AUXBUFN 00009300
//AUX1 DD DUMMY,DCB=(RECFM=VB,BLKSIZE=1004,LRECL=1000,BUFNO=&AUXBUFN) 00009400
//AUX2 DD DUMMY,DCB=(RECFM=VB,BLKSIZE=1004,LRECL=1000,BUFNO=&AUXBUFN) 00009500
//AUX3 DD DUMMY,DCB=(RECFM=VB,BLKSIZE=1004,LRECL=1000,BUFNO=&AUXBUFN) 00009600
//AUX4 DD DUMMY,DCB=(RECFM=VB,BLKSIZE=1004,LRECL=1000,BUFNO=&AUXBUFN) 00009700
//AUX5 DD DUMMY,DCB=(RECFM=VB,BLKSIZE=1004,LRECL=1000,BUFNO=&AUXBUFN) 00009800
//FMTAUX DD UNIT=&STG,SPACE=(TRK,(10,5)),DCB=BUFNO=&AUXBUFN 00009900
//SLIB DD DSNAME=*.SYSLMOD,DISP=(OLD,PASS),VOLUME=REF=*.SYSLMOD 00010000
// DD DSNAME=&LIB.L,UNIT=&ULIB,VOLUME=&VLIB,DISP=SHR 00010100
// DD DSNAME=&LIB1.L,UNIT=&ULIB1,VOLUME=&VLIB1,DISP=SHR 00010200
// DD DSNAME=&JOBLIB,DISP=SHR 00010300
//SOURCLIB DD DISP=&SDISP,DSN=&SOURCL.L,VCL=&VSCURCL,UNIT=&USOURCL 00010400
//OMMACS DD UNIT=&STG,SPACE=(CYL,(5,1)) 00010500
//GENFILE DD UNIT=&STG,SPACE=(CYL,(3,1)) 00010600
//LITFILE DD UNIT=&STG,SPACE=(CYL,(3,1)) 00010700
//EDITFILE DD UNIT=&STG,SPACE=(CYL,(3,1)) 00010800
//FMFLUD DD UNIT=&STG,SPACE=(CYL,(5,1)) 00010900
//FMTEFILE DD UNIT=&STG,SPACE=(TRK,(5,1)) 00011000
//FMAEFILE DD UNIT=&STG,SPACE=(TRK,(5,1)) 00011100
//FMSODANQ DD DISP=SHR,DSNAME=&ISAM.D,VOLUME=REF=*.STEPLIB 00011200
//NFLERR DD DSNAME=*.SORTWK04,VOLUME=REF=*.SORTWK04,DISP=(MOD,PASS) 00011300
//NFLMACS DD DSNAME=*.SORTWK03,VOLUME=REF=*.SORTWK03,DISP=(OLD,PASS) 00011400

```

## NIPS 360 FFS

## PROCEDURES

```

//ASSEMIN DD DSN=*.FMLITFIL,VOLUME=REF=*.FMLITFIL,DISP=(OLD,PASS) 00011500
// DD DSN=*.SORTWK06,VOLUME=REF=*.SORTWK06,DISP=(OLD,PASS) 00011600
//ISAMWORK DD UNIT=&STG,DCB=DSORG=IS,SPACE=(CYL,(10)) 00011700
//SAMFILE DD DSN=&SAM.S,UNIT=&USAM,VOLUME=&VSAM,DISP=(SHR,KEEP), X00011800
// LABEL=(,&LAB),DCB=(RECFM=VB,LRECL=1000,BLKSIZE=&BSZFILE,X00011900
// TRTCH=&TRCH,DEN=&DEN) 00012000
//FMSAMCUT DD &SAMOUT.DCB=(RECFM=VB,LRECL=1000,BLKSIZE=&BSZNEW, X00012100
// DEN=&DEN,TRTCH=&TRCH),DISP=(NEW,KEEP), 00012200
// LABEL=(,&LAB), 00012300
// UNIT=&USAM,VOLUME=(PRIVATE,RETAIN,&VSMCUT) 00012400
//FMNDATA DD &SAMOUT.DCB=*.FMSAMCUT,DISP=(,KEEP),LABEL=(,&LAB), X00012500
// UNIT=&USAM,VOLUME=(PRIVATE,RETAIN,&VSMCUT) 00012600
//STAT DD DSN=*.FMCM,VOL=REF=*.FMCM,DISP=(OLD,PASS) 00012700
//TRANST DD DSN=&TRANS,VOL=&VTRANS,UNIT=&UTRANS,DISP=&TDISP, 00012800
// SPACE=(TRK,1) 00012900
//INDEXPRT DD SYSOUT=(&A,&CL) 00013000
//IXTRANS DD DISP=(NEW,PASS),UNIT=&STG,SPACE=(CYL,(&INXSP,1)) 00013100
//XINDEX DD DSN=&XINDEX.X,DISP=(&XDISP,KEEP),UNIT=&XUNIT,VOL=&XVOL, 00013200
// DCB=(BLKSIZE=&BLKSIZE,RECFM=F,KEYLEN=4,DSORG=DA), X00013300
// SPACE=(&BLKSIZE,&NBRBLK) 00013400

```

## NIPS 360 FFS

## PROCEDURES

```

//XFMEX  PROC  A=A,AUX=FN=2,AUXSP=6,B=B,BLKSIZE=560,          00000100
//          BSZFILE=,BSZNEW=,                                X00000200
//          CHKDSP=DELETE,CHKID='&&CHKID',CHKSP=0,             00000300
//          CHKST=NEW,CL=' ',CL1=' ',CL2=' ',DEN=,EROPT=ABE,    00000400
//          INXSP=1,ISAM='DUMMY.FILE',JOBLIB='FFS.JOBLIB',      00000500
//          LAB=SL,LIB='DUMMY.FILE',LIB1='DUMMY.FILE',          X00000600
//          MCTSP=5,NBRBLK=200,NRMDSP=DELETE,RGN=98K,           00000700
//          PTFJOBL='PTF.JOBLIB',                               00000710
//          SAM='DUMMY.FILE',SAMOUT='DUMMY',SORTSP=10,          00000800
//          STG=NIPW,TDISP=MOD,TRANS='&&TRANS',                 00000900
//          TRANSP=200,TRCH=,UCHK=NIPW,UISAM='(2314,P)',        00001000
//          ULIB=2314,ULIB1=2314,USAM='(TAPE9,,DEFER)',          00001100
//          UTRANS=NIPW,VCHK=,VISAM=,VLIB=,VLIB1=,              00001200
//          VSAM=,VSMCT=,VSMOUT=,VTRANS=,                       00001300
//          XDISP=SHR,XINDEX='DUMMY.FILE',XUNIT=2314,XVOL=      00001400
//          **                                                    00001500
//          ** CHARLES W. HICKISCH MAJOR,USA  PRCJ CODE=763NIPS  BRANCH=431 00001600
//          ** DATE=MARCH 1, 1974                                00001700
//          **                                                    00001800
//FM      EXEC  PGM=FM,REGION=&RGN                               00001900
//STEPLIB DD DSN=&PTFJOBL,DISP=SHR                               00002000
//          DD DSN=&JOBLIB,DISP=SHR                             00002010
//FMSTAT  DD SYSOUT=(&A,&CL)                                    00002100
//SYSOUT DD SYSOUT=(&A,&CL)                                    00002200
//SYSUDUMP DD SYSOUT=(&A,&CL1)                                  00002300
//SYSPRINT DD SYSOUT=(&A,&CL2)                                  00002400
//SOURCPR DD SYSOUT=(&A,&CL)                                    00002500
//DATAFILE DD DSN=&ISAM,UNIT=&UISAM,VOLUME=&VISAM,DISP=SHR      00002600
//FMCOMM  DD UNIT=&STG,SPACE=(TRK,(1,1)),                       X00002700
//          DISP=(,&NRMDSP,&CHKDSP),DSNAME=&CHKID.A             00002800
//FMFLUD  DD UNIT=&STG,SPACE=(CYL,(&MCTSP,5)),                  X00002900
//          DISP=(,&NRMDSP,&CHKDSP),DSNAME=&CHKID.B             00003000
//FMTRANS DD UNIT=&STG,SPACE=(CYL,(&AUXSP,5)),                  X00003100
//          DISP=(,&NRMDSP,&CHKDSP),DSNAME=&CHKID.C             00003200
//SYSLMOD DD LABEL=EXPDT=66366,SPACE=(CYL,(4,1,10)),DCB=SYS1.LINKLIB, X00003300
//          UNIT=&STG,DSNAME=&CHKID.D,DISP=(,&NRMDSP,&CHKDSP)    00003400
//SORTLIB DD DSN=&SYS1.SORTLIB,DISP=SHR                          00003500
//SORTIN  DD UNIT=&STG,SPACE=(TRK,(&TRANSP,150)),              X00003600
//          DISP=(,&NRMDSP,&CHKDSP),DSNAME=&CHKID.E             00003700
//SORTWK01 DD SPACE=(CYL,(&SORTSP),,CONTIG),UNIT=(&STG,SEP=Sortin), X00003800
//          DISP=(,&NRMDSP,&CHKDSP),DSNAME=&CHKID.F             00003900
//SORTWK02 DD SPACE=(CYL,(&SORTSP),,CONTIG),                   X00004000
//          DISP=(,&NRMDSP,&CHKDSP),DSNAME=&CHKID.G,             X00004100
//          UNIT=(&STG,SEP=SortWK01),SEP=SortWK01              00004200
//SORTWK03 DD SPACE=(CYL,(&SORTSP),,CONTIG),                   X00004300
//          DISP=(,&NRMDSP,&CHKDSP),DSNAME=&CHKID.H,             X00004400
//          UNIT=(&STG,SEP=SortWK02),SEP=SortWK02              00004500
//SORTWK04 DD SPACE=(CYL,(&SORTSP),,CONTIG),                   X00004600
//          DISP=(,&NRMDSP,&CHKDSP),DSNAME=&CHKID.I,             X00004700
//          UNIT=(&STG,SEP=(SortWK01,SortWK03)),                X00004800
//          SEP=(SortWK01,SortWK03)                             00004900
//SORTWK05 DD SPACE=(CYL,(&SORTSP),,CONTIG),                   X00005000
//          DISP=(,&NRMDSP,&CHKDSP),DSNAME=&CHKID.J,             X00005100
//          UNIT=(&STG,SEP=(SortWK02,SortWK04)),                X00005200
//          SEP=(SortWK02,SortWK04)                             00005300
//SORTWK06 DD SPACE=(CYL,(&SORTSP),,CONTIG),                   00005400
//          DISP=(,&NRMDSP,&CHKDSP),DSNAME=&CHKID.K,             00005500

```



```

//          UNIT=(&STG,SEP=(SORTWK01,SORTWK03,SORTWK05)),          00005600
//          SEP=(SORTWK01,SORTWK03,SORTWK05)                        00005700
//SORTOUT   DD DISP=(OLD,PASS),DSNAME=*.SORTIN,VOLUME=REF=*.SORTIN, *00005800
//          DCB=(LRECL=1500,BLKSIZE=1504,RECFM=VB)                  00005900
//RECSIN DD DSNAME=*.FMFLUD,VOLUME=REF=*.FMFLUD,DISP=(OLD,PASS)    00006000
//RECSOUT DD DSNAME=*.RECSIN,VOLUME=REF=*.RECSIN,DISP=(OLD,PASS),  X00006100
//          DCB=(LRECL=1000,BLKSIZE=1004,RECFM=VB)                  00006200
//RECSWK01 DD DSNAME=*.SORTWK01,VOLUME=REF=*.SORTWK01,DISP=(OLD,PASS) 00006300
//RECSWK02 DD DSNAME=*.SORTWK02,VOLUME=REF=*.SORTWK02,DISP=(OLD,PASS) 00006400
//RECSWK03 DD DSNAME=*.SORTWK03,VOLUME=REF=*.SORTWK03,DISP=(OLD,PASS) 00006500
//RECSWK04 DD DSNAME=*.SORTWK04,VOLUME=REF=*.SORTWK04,DISP=(OLD,PASS) 00006600
//RECSWK05 DD DSNAME=*.SORTWK05,VOLUME=REF=*.SORTWK05,DISP=(OLD,PASS) 00006700
//RECSWK06 DD DSNAME=*.SORTWK06,VOLUME=REF=*.SORTWK06,DISP=(OLD,PASS) 00006800
//TRANS    DD DUMMY,DISP=CLD,DCB=EROPT=&EROPT                      00006900
//FMSETTBL DD UNIT=&STG,SPACE=(TRK,(1,1)),                          X00007000
//          DISP=(,ENRMDSP,&CHKDSP),DSNAME=&CHKID.L                 00007100
//FMAUXOP DD UNIT=&STG,SPACE=(CYL,(&AUXSP,5)),                      X00007200
//          DCB=(LRECL=1000,BLKSIZE=1004,BUFNO=&AUXBUFN),           00007300
//          DISP=(,ENRMDSP,&CHKDSP),DSNAME=&CHKID.M                 00007400
//PUNCH     DD SYSOUT=&B,DCB=BUFNO=&AUXBUFN                          00007500
//AUX1      DD DUMMY,DCB=(RECFM=VB,BLKSIZE=1004,LRECL=1000,BUFNO=&AUXBUFN) 00007600
//AUX2      DD DUMMY,DCB=(RECFM=VB,BLKSIZE=1004,LRECL=1000,BUFNO=&AUXBUFN) 00007700
//AUX3      DD DUMMY,DCB=(RECFM=VB,BLKSIZE=1004,LRECL=1000,BUFNO=&AUXBUFN) 00007800
//AUX4      DD DUMMY,DCB=(RECFM=VB,BLKSIZE=1004,LRECL=1000,BUFNO=&AUXBUFN) 00007900
//AUX5      DD DUMMY,DCB=(RECFM=VB,BLKSIZE=1004,LRECL=1000,BUFNO=&AUXBUFN) 00008000
//FMATAUX   DD UNIT=&STG,SPACE=(TRK,(10,5)),                      X00008100
//          DCB=BUFNO=&AUXBUFN,                                      00008200
//          DISP=(,ENRMDSP,&CHKDSP),DSNAME=&CHKID.N                 00008300
//SLIB      DD DSNAME=*.SYSLMOD,DISP=(OLD,PASS),VOLUME=REF=*.SYSLMOD 00008400
//          DD DSNAME=&LIB.L,UNIT=&ULIB,VOLUME=&VLIB,DISP=SHR         00008500
//          DD DSNAME=&LIB1.L,UNIT=&ULIB1,VOLUME=&VLIB1,DISP=SHR      00008600
//          DD DSNAME=&JOBLIB,DISP=SHR                               00008700
//FMSODANQ  DD DISP=SHR,DSNAME=&ISAM.D,VOLUME=REF=*.STEPLIB         00008800
//ISAMWORK  DD UNIT=&STG,DCB=DSORG=IS,SPACE=(CYL,(10)),            X00008900
//          DISP=(,ENRMDSP,&CHKDSP),DSNAME=&CHKID.O                 00009000
//SAMFILE   DD DSNAME=&SAM.S,UNIT=&USAM,VOLUME=&VSAM,DISP=(SHR,KEEP), X00009100
//          LABEL=(,&LAB),DCB=(RECFM=VB,LRECL=1000,BLKSIZE=&BSZFILE,X00009200
//          TPTCH=&TRCH,DEN=&DEN)                                     00009300
//FMSAMOUT  DD &SAMOUT.DCB=(RECFM=VB,LRECL=1000,BLKSIZE=&BSZNEWF, X00009400
//          DEN=&DEN,TRTCH=&TRCH),DISP=(NEW,KEEP),                  00009500
//          LABEL=(,&LAB),                                           00009600
//          UNIT=&USAM,VOLUME=(PRIVATE,RETAIN,&VSMOUT)              00009700
//FMNDATA   DD &SAMOUT.DCB=*.FMSAMOUT,DISP=(,KEEP),LABEL=(,&LAB), X00009800
//          UNIT=&USAM,VOLUME=(PRIVATE,RETAIN,&VSMOUT)              00009900
//CHECKDD   DD DSNAME=&CHKID,DISP=(&CHKST,&ENRMDSP,&CHKDSP),UNIT=&UCHK, 00010000
//          VOLUME=&VCHK,SPACE=(CYL,(&CHKSP,1))                     00010100
//STAT      DD UNIT=&STG,SPACE=(TRK,(20,5))                         00010200
//TRANST    DD DSN=&TRANS,VOL=&VTRANS,UNIT=&UTRANS,DISP=&TDISP,      00010300
//          SPACE=(TRK,1)                                           00010400
//INDEXPR   DD SYSOUT=(&A,&CL)                                       00010500
//IXTRANS   DD DISP=(NEW,PASS),UNIT=&STG,SPACE=(CYL,(&INXSP,1))    00010600
//XINDEX    DD DSN=&XINDEX.X,DISP=(&XDISP,KEEP),UNIT=&XUNIT,VOL=&XVOL, 00010700
//          DCB=(BLKSIZE=&BLKSIZE,RECFM=F,KEYLEN=4,DSORG=DA),      X00010800
//          SPACE=(&BLKSIZE,&NBRBLK)                                00010900

```

## NIPS 360 FFS

## PROCEDURES

```

//XFR      PROC  A=A,AUXSP=6,B=B,BLKSIZE=560,BSZFILE=,BSZNEW=,      X00000100
//          CL=',' ,CL1=',' ,CL2=',' ,      X00000200
//          DEN=,INXSP=1,ISAM='DUMMY.FILE',      00000300
//          JOBLIB='FFS.JOBLIB',JOBMAC='FFS.JOBMACRO',      00000400
//          LAB=SL,LIB='DUMMY.FILE',LIB1='DUMMY.FILE',      X00000500
//          MCTSP=8,NEWFFT='DUMMY.FILE',NBRBLK=200,      00000600
//          NEWUSAM='(TAPE9,,DEFER)',      00000700
//          PTFJOBL='PTF.JOBLIB',PTFJOBM='PTF.JOBMACRO',      00000710
//          RGN=100K,SAM='DUMMY.FILE',SORTSP=10,STG=NIPW,      00000800
//          TDISP=MOD,TRANS='&&TRANS',TRANSP=200,TRANTYP=ISAM,      00000900
//          TRCH=,UFFT='(2314,P)',UISAM='(2314,P)',      00001000
//          ULIB=2314,ULIB1=2314,      00001100
//          USAM='(TAPE9,,DEFER)',UTRANS=NIPW,VFFT=,VISAM=,      00001200
//          VLIB=,VLIB1=,VSAM=,VSMCT=,VSMOUT=,VSORT='REF=*.SORTIN',      00001300
//          VTRANS=,XDISP=SHR,XINDEX='DUMMY.FILE',      00001400
//          XUNIT=2314,XVOL=      00001500
//          *      00001600
//          ** CHARLES W. HICKISCH MAJOR,USA PROJ CODE=763NIPS BRANCH=431      00001700
//          ** DATE=MARCH 1, 1974      00001800
//          *      00001900
//          ** FR EXEC PGM=FR,REGION=&RGN      00002000
//          **STEPLIB DD DSN=&PTFJOBL,DISP=SHR      00002100
//          ** DD DSN=&JOBLIB,DISP=SHR      00002110
//          **SYSLIST DD SYSOUT=(&A,&CL)      00002200
//          **SYSOUT DD SYSOUT=(&A,&CL)      00002300
//          **SYSUDUMP DD SYSOUT=(&A,&CL1)      00002400
//          **SYSPRINT DD SYSOUT=(&A,&CL2)      00002500
//          **SYSPUNCH DD SYSOUT=&B      00002600
//          **OLDFILE DD DSNNAME=&ISAM,UNIT=&UISAM,VOLUME=&VISAM,DISP=SHR      00002700
//          **DATAFILE DD DSNNAME=&NEWFFT,UNIT=&UFFT,VOLUME=&VFFT,DISP=SHR      00002800
//          **SAMFILE DD DSNNAME=&SAM.S,UNIT=&USAM,VOLUME=&VSAM,LABEL=(,&LAB),      C00002900
//          ** DCB=(RECFM=VB,LRECL=1000,BLKSIZE=&BSZFILE,DEN=&DEN,      X00003000
//          ** TRCH=&TRCH),DISP=(SHR,PASS)      00003100
//          **TODRCD DD UNIT=&STG,SPACE=(CYL,(3,1))      00003200
//          **POOLRCD DD UNIT=&STG,SPACE=(CYL,(3,1))      00003300
//          **FRVSNA DD UNIT=&STG,SPACE=(CYL,(5,1))      00003400
//          **FRVSNB DD UNIT=&STG,SPACE=(CYL,(5,1))      00003500
//          **FRGENCD DD UNIT=&STG,SPACE=(TRK,1)      00003600
//          **SORTLIB DD DSNNAME=SYS1.SORTLIB,DISP=SHR      00003700
//          **FMCOMM DD UNIT=&STG,SPACE=(TRK,(1,1))      00003800
//          **FMFLUD DD UNIT=&STG,SPACE=(CYL,(&MCTSP,10))      00003900
//          **FMTRANS DD UNIT=&STG,SPACE=(CYL,(&AUXSP,5))      00004000
//          **FMLABELS DD UNIT=&STG,SPACE=(CYL,(1,1))      00004100
//          **FMCM DD UNIT=&STG,SPACE=(TRK,(20,5))      00004200
//          **FMCMFILE DD UNIT=&STG,SPACE=(TRK,(20,5))      00004300
//          **FMLITFIL DD UNIT=&STG,SPACE=(TRK,(20,5))      00004400
//          **FMLEFILE DD UNIT=&STG,SPACE=(CYL,(5,2))      00004500
//          **FMMACRO DD UNIT=&STG,SPACE=(CYL,(8,2))      00004600
//          **FMOUTMAC DD UNIT=&STG,SPACE=(CYL,(8,2))      00004700
//          **SYSLIB DD DSN=&PTFJOBM,DISP=SHR      00004800
//          ** DD DSN=&JOBMAC,DISP=SHR      00004810
//          ** DD DSNNAME=SYS1.MACLIB,DISP=SHR      00004900
//          **FMLEIN DD UNIT=&STG,SPACE=(TRK,(25,5))      00005000
//          **SYSLMOD DD UNIT=&STG,SPACE=(CYL,(4,1,10)),DCB=&JOBLIB,      C00005100
//          ** LABEL=EXPDT=66366      00005200
//          **TEMLSREC DD UNIT=&STG,SPACE=(TRK,(20,5))      00005300
//          **SORTIN DD UNIT=&STG,SPACE=(TRK,(&TRANSP,20))      00005400

```

## NIPS 360 FFS

## PROCEDURES

```

//SORTWK01 DD SPACE=(CYL,(&SORTSP),,CONTIG),UNIT=&STG 00005500
//SORTWK02 DD SPACE=(CYL,(&SORTSP),,CONTIG),UNIT=(&STG,SEP=) 00005600
//SORTWK03 DD SPACE=(CYL,(&SORTSP),,CONTIG),UNIT=(&STG,SEP=) 00005700
//SORTWK04 DD SPACE=(CYL,(&SORTSP),,CONTIG), 00005800
// UNIT=(&STG,SEP=(SORTWK01,SORTWK03)) 00005900
//SORTWK05 DD SPACE=(CYL,(&SORTSP),,CONTIG), 00006000
// UNIT=(&STG,SEP=(SORTWK02,SORTWK04)) 00006100
//SORTWK06 DD SPACE=(CYL,(&SORTSP),,CONTIG), 00006200
// UNIT=(&STG,SEP=(SORTWK03,SORTWK05)) 00006300
//SORTOUT DD DSN=*.SORTIN,VOLUME=REF=*.SORTIN,DISP=(OLD,PASS), 00006400
// DCB=(LRECL=1500,BLKSIZE=1504,RECFM=VB) 00006500
//TAPEIN DD DSN=*.SORTIN,VOLUME=&VSORT,DISP=(OLD,PASS) 00006600
//TAPEOUT DD DSN=*.SORTIN,VOLUME=&VSORT,DISP=(OLD,PASS), 00006700
// DCB=(LRECL=1500,BLKSIZE=1504,RECFM=VB) 00006800
//RECSIN DD DSN=*.FMFLUD,VOLUME=REF=*.FMFLUD,DISP=(OLD,PASS) 00006900
//RECSOUT DD DSN=*.RECSIN,VOLUME=REF=*.RECSIN,DISP=(OLD,PASS), 00007000
// DCB=(LRECL=1000,BLKSIZE=1004,RECFM=VB) 00007100
//RECSWK01 DD DSN=*.SORTWK01,VOLUME=REF=*.SORTWK01,DISP=(OLD,PASS) 00007200
//RECSWK02 DD DSN=*.SORTWK02,VOLUME=REF=*.SORTWK02,DISP=(OLD,PASS) 00007300
//RECSWK03 DD DSN=*.SORTWK03,VOLUME=REF=*.SORTWK03,DISP=(OLD,PASS) 00007400
//RECSWK04 DD DSN=*.SORTWK04,VOLUME=REF=*.SORTWK04,DISP=(OLD,PASS) 00007500
//RECSWK05 DD DSN=*.SORTWK05,VOLUME=REF=*.SORTWK05,DISP=(OLD,PASS) 00007600
//RECSWK06 DD DSN=*.SORTWK06,VOLUME=REF=*.SORTWK06,DISP=(OLD,PASS) 00007700
//TRANS DD DSN=*.TRANS,UNIT=&TRANS,VOLUME=&TRANS,DISP=SHR 00007800
//ISAM DD DSN=*.ISAM,UNIT=&ISAM,VOLUME=&ISAM,DISP=SHR 00007900
//SAM DD DSN=*.SAM,UNIT=&SAM,VOLUME=&SAM,LABEL=(,LAB), 00008000
// DCB=*.SAMFILE,DISP=(SHR,KEEP) 00008100
//ASSEMIN DD DSN=*.FMOUTMAC,VOLUME=REF=*.FMOUTMAC,DISP=(OLD,PASS) 00008200
//FMSETTBL DD UNIT=&STG,SPACE=(TRK,(1,1)) 00008300
//FMAUXOP DD DSN=*.FMTRANS,VOLUME=REF=*.FMTRANS,DISP=(OLD,PASS), 00008400
// DCB=(LRECL=1000,BLKSIZE=1004) 00008500
//SLIB DD DSN=*.SYSLMOD,DISP=(OLD,PASS),VOLUME=REF=*.SYSLMOD 00008600
// DD DSN=*.LIB.L,UNIT=&LIB,VOLUME=&LIB,DISP=SHR 00008700
// DD DSN=*.LIB1.L,UNIT=&LIB1,VOLUME=&LIB1,DISP=SHR 00008800
// DD DSN=*.LIB2.L,DISP=SHR 00008900
//FMSAMOUT DD UNIT=&NEWUSAM,VOLUME=(PRIVATE,RETAIN,&VSMOUT), 00009000
// LABEL=(,LAB),DISP=(NEW,KEEP), 00009100
// DCB=(RECFM=VB,LRECL=1000,BLKSIZE=&BSZNEW,DEN=&DEN, 00009200
// TRTCH=&TRCH) 00009300
//FMNDATA DD UNIT=&NEWUSAM,VOLUME=(PRIVATE,RETAIN,&VSMOUT), 00009400
// LABEL=(,LAB),DISP=(,KEEP),DCB=*.FMSAMOUT 00009500
//STAT DD DSN=*.FMCM,VOL=REF=*.FMCM,DISP=(OLD,PASS) 00009600
//TRANST DD DSN=&TRANS,VOL=&TRANS,UNIT=&TRANS,DISP=&TDISP, 00009700
// SPACE=(TRK,1) 00009800
//INDEXPRT DD SYSOUT=(,CL) 00009900
//IXTRANS DD DISP=(NEW,PASS),UNIT=&STG,SPACE=(CYL,(&INXSP,1)) 00010000
//XINDEX DD DSN=&INDEX.X,DISP=(,DISP,KEEP),UNIT=&XUNIT,VOL=&XVOL, 00010100
// DCB=(BLKSIZE=&BLKSIZE,RECFM=F,KEYLEN=4,DSORG=DA), 00010200
// SPACE=(&BLKSIZE,&NBRBLK) 00010300

```



## NIPS 360 FFS

## PROCEDURES

```

//XFS      PROC  A=A,CL=*,*,CL1=*,*,BSZNEWF=1004,INDEX=1,      X00000100
//          ISAM='DUMMY.FILE',      X00000200
//          JOBLIB='FFS.JOBLIB',      00000300
//          LIB='DUMMY.FILE',LIB1='DUMMY.FILE',      00000400
//          NDISP=KEEP,OVLW=1,PRIME=5,RGN=60K,SDISP=SHR,      00000500
//          PTFJOBL='PTF.JOBLIB',      00000510
//          SORTSP=3,SOURCL='DUMMY.FILE',STG=NIPW,      00000600
//          UISAM=2314,ULIB=2314,ULIB1=2314,USOURCL=2314,      00000700
//          VISAM=,VLIB=,VLIB1=,VSOURCL=      00000800
//          00000900
// * CHARLES W. HICKISCH MAJOR,USA PROJ CODE=763NIPS BRANCH=431      00001000
// * DATE=MARCH 1, 1974      00001100
// *      00001200
//FS      EXEC  PGM=FSPHASE,REGION=&RGN      00001300
//STEPLIB DD DSN=&PTFJOBL,DISP=SHR      00001400
//          DD DSN=&JOBLIB,DISP=SHR      00001410
//SLIB    DD DSN=&LIB.L,DISP=SHR,VOLUME=&VLIB,UNIT=&ULIB      00001500
//          DD DSN=&LIB1.L,UNIT=&ULIB1,VOLUME=&VLIB1,DISP=SHR      00001600
//          DD DSN=&JOBLIB,DISP=SHR      00001700
//SOURCLIB DD DISP=&SDISP,DSN=&SOURCL.L,VOL=&VSOURCL,UNIT=&USOURCL      00001800
//NEWFILE DD DSN=&ISAM.(INDEX),UNIT=&UISAM,VOLUME=&VISAM,      *00001900
//          SPACE=(CYL,&INDEX),DCB=(DSCRG=IS,BLKSIZE=&BSZNEWF),      X00002000
//          DISP=(,ANDISP,DELETE)      00002100
//          DD DSN=&ISAM.(PRIME),UNIT=&UISAM,VOLUME=REF=*.NEWFILE,      *00002200
//          SPACE=(CYL,&PRIME),DCB=*.NEWFILE,DISP=(,ANDISP,DELETE)      00002300
//          DD DSN=&ISAM.(OVFLOW),UNIT=&UISAM,VOLUME=REF=*.NEWFILE,      *00002400
//          SPACE=(CYL,&OVFLOW),DCB=*.NEWFILE,DISP=(,ANDISP,DELETE)      00002500
//SORTLIB DD DSN=SYS1.SORTLIB,DISP=SHR      00002600
//SORTIN DD UNIT=&STG,SPACE=(CYL,(1,1)),      *00002700
//          DCB=(RECFM=VB,BLKSIZE=1004,LRECL=1000)      00002800
//SORTWK01 DD SPACE=(CYL,(&SORTSP),,CONTIG),UNIT=(&STG,SEP=sortin)      00002900
//SORTWK02 DD SPACE=(CYL,(&SORTSP),,CONTIG),UNIT=(&STG,SEP=sortwk01),      *00003000
//          SEP=sortwk01      00003100
//SORTWK03 DD SPACE=(CYL,(&SORTSP),,CONTIG),UNIT=(&STG,SEP=sortwk02),      *00003200
//          SEP=sortwk02      00003300
//SORTWK04 DD SPACE=(CYL,(&SORTSP),,CONTIG),      *00003400
//          UNIT=(&STG,SEP=(sortwk01,sortwk03)),      *00003500
//          SEP=(sortwk01,sortwk03)      00003600
//SORTWK05 DD SPACE=(CYL,(&SORTSP),,CONTIG),      *00003700
//          UNIT=(&STG,SEP=(sortwk02,sortwk04)),      *00003800
//          SEP=(sortwk02,sortwk04)      00003900
//SORTWK06 DD SPACE=(CYL,(&SORTSP),,CONTIG),      *00004000
//          UNIT=(&STG,SEP=(sortwk01,sortwk03,sortwk05)),      *00004100
//          SEP=(sortwk01,sortwk03,sortwk05)      00004200
//SORTOUT DD UNIT=&STG,SPACE=(CYL,(1,1)),      *00004300
//          DCB=(RECFM=VB,BLKSIZE=1004,LRECL=1000)      00004400
//SYSOUT DD DUMMY      00004500
//SYSPRINT DD SYSOUT=(&A,&CL)      00004600
//SYSUDUMP DD SYSOUT=(&A,&CL1)      00004700
//SOURCEPR DD SYSOUT=(&A,&CL)      00004800
//SYSUT1 DD DISP=(,PASS),DSN=IXSPDS,UNIT=&STG,SPACE=(TRK,(5,1))      00004900
//FSSTEP2 EXEC PGM=FSABMOD,COND=(111,NE,FS),REGION=&RGN      00005000
//STEPLIB DD DSN=&JOBLIB,DISP=SHR      00005100
//SYSOUT DD SYSOUT=(&A,&CL)      00005200
//SYSPRINT DD SYSOUT=(&A,&CL)      00005300
//DELNEW DD DSNNAME=*.FS.NEWFILE,UNIT=&UISAM,VOLUME=REF=*.FS.NEWFILE, *00005400
//          DISP=(OLD,DELETE)      00005500

```

## NIPS 360 FFS

## PROCEDURES

```

//XISTOS PROC  A=A,BSZFILE=,BSZNEWF=,CC=,CL=',' ,CL1=',' ,DEN=,          00000100
//              I SAM='DUMMY.FILE',JOB LIB='FFS.JOB LIB',LAB=SL,NDISP=KEEP, 00000200
//              ODISP=KEEP,OLDSAM='DUMMY.FILE',CLDVSAM=,          00000300
//              OSDISP=KEEP,PGN=60K,SAM='DUMMY.FILE',SEQNC=1,      00000400
//              PTFJOB LIB='PTF.JOB LIB',          00000410
//              TRCH=,UISAM='(2314,P)',USAM='(TAPE9,,DEFER)',      00000500
//              VISAM=,VSAM=          00000600
//              00000700
// * CHARLES W. HICKISCH MAJOR,USA PROJ CODE=763NIPS BRANCH=431 00000800
// * DATE=MARCH 1, 1974          00000900
// *          00001000
//STPL EXEC PGM=UTBLDSAM,REGION=&RG,N,PARM='&CC'          00001100
//STEPLIB DD DSN=&PTFJOB LIB,DISP=SHR          00001200
//          DD DSN=&JOB LIB,DISP=SHR          00001210
//DATAFILE DD DSNAME=&I SAM,UNIT=&UISAM,VOLUME=&VISAM,      00001300
//          DISP=(SHR,&ODISP,KEEP),DCB=BUFNO=5          00001400
//SAMFILE DD DSNAME=&OLDSAM.S,UNIT=&USAM,DISP=(SHR,&CSDISP), 00001500
//          VOL=&OLDVSAM,DCB=(RECFM=VB,LRECL=1000,BLKSIZE=&BSZFILE, 00001600
//          DEN=&DEN,TRTCH=&TRCH,BUFNO=5),LABEL=(&SEQNC,&LAB) 00001700
//SAMOUT DD DSNAME=&SAM.S,UNIT=&USAM,VOLUME=&VSAM,DISP=(, &NDISP), 00001800
//          DCB=(*,SAMFILE,BLKSIZE=&BSZNEWF),LABEL=(&SEQNO,&LAB) 00001900
//SYSPRINT DD SYSOUT=(&A,&CL)          00002000
//SYSUDUMP DD SYSOUT=(&A,&CL1)          00002100

```



## NIPS 360 FFS

## PROCEDURES

```

//XKA  PROC  A=A,CL=',',CL1=',',JOB LIB='FFS.JOB LIB',RGN=100K, 00000100
//          PTFJOBL='PTF.JOB LIB', 00000110
//          STG=NIPW, SORTSP=10, 00000200
//          LIB=,ULIB=2314,VLIB=, 00000300
//          ISAM='DUMMY.FILE',UISAM='(2314,P)',VISAM=, 00000400
//          SAM='DUMMY.FILE',USAM='(TAPE9,,DEFER)',VSAM=, 00000500
//          LAB=SL,DEN=3,TRCH= 00000600
//          00000700
//          PARAMETERS 00000800
//          00000900
//          00001000
//          A          SYSOUT - DEFAULT=PRINTER 00001100
//          CL          SYSOUT CLASSIFICATION - REQUIRED 00001200
//          CL1          SYSUDUMP CLASSIFICATION - REQUIRED 00001300
//          JOBLIB       STEPLIB DSN - DEFAULT=FFS.JOB LIB 00001400
//          RGN          STEP REGION - DEFAULT=175K 00001500
//          PTFJOBL      STEPLIB DSN - DEFAULT=PTF.JOB LIB 00001600
//          STG          TEMP STORAGE UNIT - DEFAULT=NIPW 00001700
//          SORTSP       SORTWKNN CYLINDERS - DEFAULT=10 00001800
//          LIB          USER LIBRARY DSN - REQUIRED 00001900
//          ULIB         USER LIBRARY UNIT - DEFAULT=2314 0002000
//          VLIB         USER LIBRARY 'SER=SERIAL' - REQUIRED 00002100
//          00002200
//          ISAM        DSNNAME IF ISAM DATA FILE. OMIT IF SAM FILE. 00002300
//          UISAM        ISAM UNIT CLASS 00002400
//          VISAM        ISAM VOL SERIAL 00002500
//          00002600
//          SAM         DSNNAME IF SAM DATA FILE. OMIT IF ISAM FILE. 00002700
//          USAM         SAM UNIT CLASS 00002800
//          VSAM         SAM VOL SERIAL 00002900
//          LAB         SAM VOL LABEL IF NOT STANDARD 00003000
//          DEN         SAM VOL DENSITY IF NOT 1600 BPI. 00003100
//          TRCH        SAM CONVERSION IF M-TRACK VOL. 00003200
//          00003300
//          CHARLES W. HICKISCH MAJOR,USA  PRGJ CODE=763NIPS  BRANCH=431 00003400
//          DATE=MARCH 1,1974 00003500
//          00003600
//          EXEC  PGM=UTNDXKAN, 00003700
//          REGION=&RGN 00003800
//          00003900
//          STEPLIB DD DSN=&PTFJOBL, 00004000
//          DISP=SHR 00004100
//          DD DSN=&JOBLIB, 00004200
//          DISP=SHR 00004300
//          00004400
//          00004500
//          DATAFILE DD DSN=&ISAM, 00004600
//          DISP=SHR, 00004700
//          UNIT=&UISAM, 00004800
//          VOL=&VISAM 00004900
//          00005000
//          SAMFILE DD DSN=&SAM.S, 00005100
//          UNIT=&USAM, 00005200
//          VOL=&VSAM, 00005300
//          DISP=(SHR,KEEP), 00005400
//          LABEL=(,&LAB), 00005500
//          DCB=(DEN=&DEN,TRCH=&TRCH) 00005600
//          00005700

```

## NIPS 360 FFS

## PROCEDURES

//SLIB	DD	DSN=&LIB.L,	00005400
//		UNIT=&ULIB,	00005500
//		VOL=&VLIB,	00005600
//		DISP=SHR	00005700
/**			00005800
//	DD	DSN=&JOB LIB,DISP=SHR	00005900
/**			00006000
//KANWK01	DD	UNIT=&STG,	00006100
//		DISP=(,PASS),	00006200
//		SPACE=(CYL,(&SORTSP,&SCRTSP))	00006300
/**			00006400
//SYSOUT	DD	SYSOUT=(&A,&CL)	00006500
/**			00006600
//SYSPRINT	DD	SYSOUT=(&A,&CL)	00006700
/**			00006800
//KMDPRT	DD	SYSOUT=(&A,&CL),SPACE=(CYL,(2,1))	00006900
/**			00007000
//SYSUDUMP	DD	SYSOUT=(&A,&CL1)	00007100
/**			00007200
//SORTLIB	DD	DSN=SYS1.SORTLIB,	00007300
//		DISP=SHR	00007400
/**			00007500
//SORTWK01	DD	UNIT=&STG,	00007600
//		SPACE=(CYL,(&SORTSP),,CONTIG)	00007700
/**			00007800
//SORTWK02	DD	UNIT=(&STG,SEP=SORTWK01),SEP=SORTWK01,	00007900
//		SPACE=(CYL,(&SORTSP),,CONTIG)	00008000
/**			00008100
//SORTWK03	DD	UNIT=(&STG,SEP=SORTWK02),SEP=SCRTWK02,	00008200
//		SPACE=(CYL,(&SORTSP),,CONTIG)	00008300
/**			00008400
//SORTWK04	DD	UNIT=(&STG,SEP=(SORTWK01,SORTWK03)),	00008500
//		SEP=(SORTWK01,SORTWK03),	00008600
//		SPACE=(CYL,(&SORTSP),,CONTIG)	00008700
/**			00008800
//SORTWK05	DD	UNIT=(&STG,SEP=(SORTWK02,SORTWK04)),	00008900
//		SEP=(SORTWK02,SORTWK04),	00009000
//		SPACE=(CYL,(&SORTSP),,CONTIG)	00009100
/**			00009200
//SORTWK06	DD	UNIT=(&STG,SEP=(SORTWK01,SCRTWK03,SORTWK05)),	00009300
//		SEP=(SORTWK01,SORTWK03,SORTWK05),	00009400
//		SPACE=(CYL,(&SCRTSP),,CONTIG)	00009500
/**			00009600

## NIPS 360 FFS

## PROCEDURES

```

//XKM  PROC  A=A,CL=',',CL1=',',JOB LIB='FFS.JOB LIB',RGN=100K,      00000100
//          PTFJOB LIB='PTF.JOB LIB',                                00000110
//          STG=NIPW,SGRTSP=10,                                       00000200
//          LIB=,ULIB=2314,VLIB=,DLIB='MOD,KEEP',                    00000300
//          PRISP=2,SECSP=2,BLKSI=3478                                00000400
//                                                                    00000500
//*                                                                    00000600
//*          PARAMETERS                                              00000700
//* A          SYSOUT - DEFAULT=PRINTER                               00000800
//* CL         SYSOUT CLASSIFICATION - REQUIRED                       00000900
//* CL1        SYSUDUMP CLASSIFICATION - REQUIRED                     00001000
//* JOB LIB    STEPLIB DSN - DEFAULT=FFS.JOB LIB                    00001100
//* RGN        STEP REGION - DEFAULT=175K                           00001200
//* PTFJOB LIB STEPLIB DSN - DEFAULT=PTF.JOB LIB                    00001210
//* STG        TEMP STORAGE UNIT - DEFAULT=NIPW                     00001300
//* SORTSP     SORTWKNN CYLINDERS - DEFAULT=10                      00001400
//* LIB        USER LIBRARY DSN - REQUIRED                           00001500
//* ULIB       USER LIBRARY UNIT - DEFAULT=2314                    00001600
//* VLIB       USER LIBRARY 'SER=SERIAL' - REQUIRED                  00001700
//* DLIB       USER LIBRARY DISP - DEFAULT='MOD,KEEP'              00001800
//* PRISP      WORK SPACE PRIMARY CYLS - DEFAULT=2                  00001900
//* SECSP      WORK SPACE SECONDARY CYLS - DEFAULT=2                00002000
//* BLKSI      WORK SPACE DCB BLKSIZE - DEFAULT=94                  00002100
//*                                                    00002200
//* CHARLES W. HICKISCH MAJOR,USA  PRCJ CODE=763NIPS  BRANCH=431    00002300
//* DATE=MARCH 1,1974                                              00002400
//*                                                    00002500
//XKM  EXEC  PGM=UTNDXKMD,                                           00002600
//          REGION=&RGN                                              00002700
//*                                                    00002800
//STEPLIB DD DSN=&PTFJOB LIB,                                         00002900
//          DISP=SHR                                                00002910
//          DD DSN=&JOB LIB,                                           00002920
//          DISP=SHR                                                00003000
//*                                                    00003100
//SLIB DD DSN=&LIB.L,                                                 00003200
//          UNIT=&ULIB,                                              00003300
//          VOL=&VLIB,                                               00003400
//          DISP=(&DLIB)                                             00003500
//*                                                    00003600
//KMDWK01 DD UNIT=&STG,                                               00003700
//          DISP=(,PASS),                                           00003800
//          SPACE=(CYL,(&PRISP,&SECSP)),                             00003900
//          DCB=(RECFM=FBS,LRECL=94,BLKSIZE=&BLKSI)                 00004000
//*                                                    00004100
//KMDWK02 DD UNIT=&STG,                                               00004200
//          DISP=(,PASS),                                           00004300
//          SPACE=(CYL,(&PRISP,&SECSP)),                             00004400
//          DCB=(RECFM=FBS,LRECL=94,BLKSIZE=&BLKSI)                 00004500
//*                                                    00004600
//KMDWK03 DD UNIT=&STG,                                               00004700
//          DISP=(,PASS),                                           00004800
//          SPACE=(TRK,(1,1))                                         00004900
//*                                                    00005000
//KMDWK04 DD UNIT=&STG,                                               00005100
//          DISP=(,PASS),                                           00005200
//          SPACE=(TRK,(5,5)),                                         00005300

```

## NIPS 360 FFS

## PROCEDURES

```
//          DCB=(RECFM=FBS,LRECL=72,BLKSIZE=720)          00005400
//*          00005500
//KMDWK05 DD UNIT=&STG,          00005600
//          DISP=(,PASS),          00005700
//          SPACE=(CYL,(&PRISP,&SECSP)),          00005800
//          DCB=(RECFM=FBS,LRECL=94,BLKSIZE=&BLKSI)          00005900
//*          00006000
//SYSOUT DD SYSOUT=(&A,&CL)          00006100
//*          00006200
//SYSPRINT DD SYSOUT=(&A,&CL)          00006300
//KMDPRT DD SYSOUT=(&A,&CL),SPACE=(CYL,(2,1))          00006400
//*          00006500
//SYSUDUMP DD SYSOUT=(&A,&CL1)          00006600
//*          00006700
//SORTLIB DD DSN=SYS1.SORTLIB,          00006800
//          DISP=SHR          00006900
//*          00007000
//SORTWK01 DD UNIT=&STG,          00007100
//          SPACE=(CYL,(&SORTSP),,CONTIG)          00007200
//*          00007300
//SORTWK02 DD UNIT=(&STG,SEP=SCRTWK01),SEP=SCRTWK01,          00007400
//          SPACE=(CYL,(&SORTSP),,CONTIG)          00007500
//*          00007600
//SORTWK03 DD UNIT=(&STG,SEP=SCRTWK02),SEP=SCRTWK02,          00007700
//          SPACE=(CYL,(&SORTSP),,CONTIG)          00007800
//*          00007900
//SORTWK04 DD UNIT=(&STG,SEP=(SORTWK01,SORTWK03)),          00008000
//          SEP=(SORTWK01,SORTWK03),          00008100
//          SPACE=(CYL,(&SORTSP),,CONTIG)          00008200
//*          00008300
//SORTWK05 DD UNIT=(&STG,SEP=(SORTWK02,SORTWK04)),          00008400
//          SEP=(SORTWK02,SORTWK04),          00008500
//          SPACE=(CYL,(&SORTSP),,CONTIG)          00008600
//*          00008700
//SORTWK06 DD UNIT=(&STG,SEP=(SORTWK01,SCRTWK03,SORTWK05)),          00008800
//          SEP=(SORTWK01,SORTWK03,SORTWK05),          00008900
//          SPACE=(CYL,(&SORTSP),,CONTIG)          00009000
//*          00009100
```



## NIPS 360 FFS

## PROCEDURES

```

//XOP  PROC  A=A,B=BSZFILE=,BSZFIL1=,BSZFIL2=, 00000100
//          CL=',',CL1=',',CL2=',', 00000200
//          ISAM='DUMMY.FILE',ISAM1='DUMMY.FILE',ISAM2='DUMMY.FILE', 00000300
//          JOBLIB='FFS.JOBLIB',JOBMAC='FFS.JOBMACRO', 00000400
//          LAB=SL, 00000500
//          LIB='DUMMY.FILE',LIB1='DUMMY.FILE',LIB2='DUMMY.FILE', 00000600
//          LIBDISP=SHR, 00000700
//          PTFJOBL='PTF.JOBLIB', 00000710
//          PTFJOBM='PTF.JOBMACRO', 00000720
//          QDF='&&QDFILE',QDISP=PASS,QRT='&&CPTFILE', 00000800
//          RGN=100K, 00000900
//          SAM='DUMMY.FILE',SAM1='DUMMY.FILE',SAM2='DUMMY.FILE', 00001000
//          SDISP=SHR,SOURCL='DUMMY.FILE',STG=NIPW, 00001100
//          TOISP=MOD,TRANS='&&TRANS', 00001200
//          UISAM='(2314,P)', 00001300
//          ULIB=2314,ULIB1=2314,ULIB2=2314, 00001400
//          UQDF=NIPW,UQRT=NIPW, 00001500
//          USAM='(TAPE9,,DEFER)',USOURCL=2314, 00001600
//          UTRANS=NIPW, 00001700
//          VISAM=,VISAM1=,VISAM2=, 00001800
//          VLIB=,VLIB1=,VLIB2=, 00001900
//          VQDF=,VQRT=, 00002000
//          VSAM=,VSAM1=,VSAM2=, 00002100
//          VSOURCL=,VTRANS= 00002200
// 00002300
//* CHARLES W. HICKISCH MAJOR,USA PROJ CODE=763NIPS BRANCH=431 00002400
//* DATE=MARCH 1, 1974 00002500
//* 00002600
//OP EXEC PGM=OP,REGION=&RGN 00002700
//STEPLIB DD DSN=&PTFJOBL,DISP=SHR 00002800
//          DD DSN=&JOBLIB,DISP=SHR 00002810
//DATAFILE DD DSN=&ISAM,DISP=SHR,UNIT=&UISAM,VOLUME=&VISAM 00002900
//DATAFIL1 DD DSN=&ISAM1,DISP=SHR,UNIT=&UISAM,VOLUME=&VISAM1 00003000
//DATAFIL2 DD DSN=&ISAM2,DISP=SHR,UNIT=&UISAM,VOLUME=&VISAM2 00003100
//SAMFILE DD DSN=&SAM.S,DISP=SHR,UNIT=&USAM,VOLUME=&VSAM, 00003200
//          LABEL=(, &LAB),DCB=BLKSIZE=&BSZFILE 00003300
//SAMFILE1 DD DSN=&SAM1.S,DISP=SHR,UNIT=&USAM,VOLUME=&VSAM1, 00003400
//          LABEL=(, &LAB),DCB=BLKSIZE=&BSZFIL1 00003500
//SAMFILE2 DD DSN=&SAM2.S,DISP=SHR,UNIT=&USAM,VOLUME=&VSAM2, 00003600
//          LABEL=(, &LAB),DCB=BLKSIZE=&BSZFIL2 00003700
//DLIB DD DSN=&LIB.L,DISP=&LIBDISP,VOLUME=&VLIB,UNIT=&ULIB 00003800
//TLIB DD DSN=&LIB1.L,DISP=SHR,VOLUME=&VLIB1,UNIT=&ULIB1 00003900
//SOURCPRT DD SYSOUT=(&A,&CL) 00004000
//SOURCLIB DD DISP=&SDISP,DSN=&SOURCL.L,VCL=&VSOURCL,UNIT=&USOURCL 00004100
//SYSPRINT DD SYSOUT=(&A,&CL),DCB=(LRECL=133,BLKSIZE=665) 00004200
//SYSOUT DD SYSOUT=(&A,&CL) 00004300
//SYSUDUMP DD SYSOUT=(&A,&CL) 00004400
//SYSUT1 DD UNIT=&STG,SPACE=(TRK,(200,20)) 00004500
//SYSUT2 DD UNIT=(&STG,SEP=SYSUT1),SPACE=(TRK,(200,20)) 00004600
//SYSUT3 DD UNIT=(&STG,SEP=(SYSUT1,SYSUT2)),SPACE=(TRK,(200,20)) 00004700
//SYSPUNCH DD UNIT=&STG,SPACE=(TRK,(50,10)), 00004800
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=400) 00004900
//SYSLIB DD DSN=&PTFJOBM,DISP=SHR 00005000
//          DD DSN=&JOBMAC,DISP=SHR 00005010
//SYSLMOD DD UNIT=&STG,SPACE=(TRK,(50,25,5)),DCB=SYS1.LINKLIB, 00005100
//          LABEL=RETPD=0000 00005200
//SYSTPRT DD SYSOUT=(&A,&CL),DCB=(RECFM=FB,LRECL=121,BLKSIZE=605) 00005300

```

## NIPS 360 FFS

## PROCEDURES

```

//OPIWCF DD UNIT=&STG,SPACE=(TRK,(2,1)),DCB=(BLKSIZE=400,LRECL=80, *00005400
// RECFM=FB) 00005500
//CPSTST DD UNIT=&STG,SPACE=(TRK,(10,2)) 00005600
//OPCREATE DD UNIT=&STG,SPACE=(TRK,(1,1)) 00005700
//OPCOMREC DD UNIT=&STG,SPACE=(TRK,(1,1)) 00005800
//INSTS DD UNIT=&STG,SPACE=(TRK,(50,5)) 00005900
//LCF DD UNIT=&STG,SPACE=(TRK,(10,5)) 00006000
//LCT DD UNIT=&STG,SPACE=(TRK,(10,5)) 00006100
//LITSTR DD UNIT=&STG,SPACE=(TRK,(50,5)) 00006200
//NAMDEF DD DSNAME=*.SYSUT1,VOLUME=REF=*.SYSUT1,DISP=(OLD,PASS) 00006300
//LITAB DD DSNAME=*.SYSUT2,VOLUME=REF=*.SYSUT2,DISP=(CLD,PASS) 00006400
//RGTAB DD DSNAME=*.SYSUT3,VOLUME=REF=*.SYSUT3,DISP=(CLD,PASS) 00006500
//CGC DD UNIT=&STG,SPACE=(TRK,(1,1)) 00006600
//ERRTAB DD UNIT=&STG,SPACE=(TRK,(8,5)) 00006700
//DCTNY DD DSNAME=*.SYSPUNCH,VOLUME=REF=*.SYSPUNCH,DISP=(CLD,PASS) 00006800
//SUBTAB DD UNIT=&STG,SPACE=(TRK,(1,1)) 00006900
//OPLGOGO DD DSNAME=*.SYSUT1,VOLUME=REF=*.SYSUT1,DISP=(CLD,PASS) 00007000
//OPLEXEX DD DSNAME=*.SYSUT2,VOLUME=REF=*.SYSUT2,DISP=(CLD,PASS) 00007100
//OPLEXDIR DD DSNAME=*.SYSUT3,VOLUME=REF=*.SYSUT3,DISP=(CLD,PASS) 00007200
//OPLINE DD SYSOUT=(&A,&CL2),DCB=LRECL=133 00007300
//OPPUNCH DD SYSOUT=&B 00007400
//SLIB DD DSNAME=*.SYSLMOD,VOLUME=REF=*.SYSLMOD,DISP=(CLD,PASS) 00007500
// DD DSNAME=*.DLIB,DISP=SHR,UNIT=&ULIB,VOLUME=&VLIB 00007600
// DD DSNAME=*.TLIB,DISP=SHR,UNIT=&ULIB1,VOLUME=&VLIB1 00007700
// DD DSNAME=&LIB2.L,DISP=SHR,UNIT=&ULIB2,VOLUME=&VLIB2 00007800
// DD DSNAME=&JOBLIB,DISP=SHR 00007900
//SYSAIN DD DSNAME=*.LITSTR,VOLUME=REF=*.LITSTR,DISP=(OLD,PASS), 00008000
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=400,BUFNO=5) 00008100
// DD DSNAME=*.INSTS,VOLUME=REF=*.INSTS,DISP=(OLD,PASS), 00008200
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=400,BUFNO=5) 00008300
//SYSLIN DD DSNAME=*.LCF,VOLUME=REF=*.LCF,DISP=(CLD,PASS) 00008400
//SYSTIN DD DSNAME=*.LCT,VOLUME=REF=*.LCT,DISP=(OLD,PASS) 00008500
//LOADR DD DSNAME=*.SYSPUNCH,VOLUME=REF=*.SYSPUNCH,DISP=(OLD,PASS), X00008600
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=400) 00008700
//QDFILE DD DSNAME=&QDF,UNIT=&UQDF,VOLUME=&VQDF,DISP=(SHR,&QDISP,KEEP) 00008800
//QRTFILE DD DSNAME=&QRT,UNIT=&UQRT,VOLUME=&VQRT,DISP=(SHR,&QDISP,KEEP) 00008900
//STAT DD DSN=*.INSTS,VOL=REF=*.INSTS,DISP=(CLD,PASS) 00009000
//TRANST DD DSN=&TRANS,VOL=&VTRANS,UNIT=&UTRANS, 00009100
// DISP=&TDISP,SPACE=(TRK,1) 00009200

```

```

//XOPEX  PRJC  A=A,B=B,                                00000100
//              CL=',',CL1=',',CL2=',',                00000200
//              CHKDSP=DELETE,CHKID='&&CHKID',CHKSP=0,CHKST=NEW, 00000300
//              ISAM='DUMMY.FILE',ISAM1='DUMMY.FILE',ISAM2='DUMMY.FILE', 00000400
//              JOBLIB='FFS.JOBLIB',                    00000500
//              LIB='DUMMY.FILE',LIB1='DUMMY.FILE',LIB2='DUMMY.FILE', 00000600
//              LIBDISP=SHR,                              00000700
//              NRMCDSP=DELETE,                          00000800
//              PTFJOBL='PTF.JOBLIB',                    00000810
//              QCF='&&QDFILE',QDISP=PASS,QRT='&&QRTFILE', 00000900
//              RGN=100K,                                  00001000
//              STG=NIPW,                                  00001100
//              TDISP=MOD,TRANS='&&TRANS',               00001200
//              UCHK=NIPW,                                 00001300
//              UISAM='(2314,P)',                          00001400
//              ULIB=2314,ULIB1=2314,ULIB2=2314,          00001500
//              UQDF=NIPW,UQRT=NIPW,                     00001600
//              UTRANS=NIPW,                              00001700
//              VISAM=,VISAM1=,VISAM2=,                  00001800
//              VLIB=,VLIB1=,VLIB2=,                     00001900
//              VQDF=,VQRT=,                              00002000
//              VCHK=,VTRANS=                             00002100
//**                                                     00002200
//**  CHARLES W. HICKISCH  MAJOR,USA  PROJ CODE=763NIPS  BRANCH=431 00002300
//**  DATE=MARCH 1, 1974                                     00002400
//**                                                         00002500
//OP EXEC PGM=OP,REGION=ERGN                               00002600
//STEPLIB DD DSN=&PTFJOBL,DISP=SHR                        00002700
//          DD DSN=&JOBLIB,DISP=SHR                       00002710
//DATAFILE DD DSN=&ISAM,DISP=SHR,UNIT=&UISAM,VOLUME=&VISAM 00002800
//DATAFIL1 DD DSN=&ISAM1,DISP=SHR,UNIT=&UISAM,VOLUME=&VISAM1 00002900
//DATAFIL2 DD DSN=&ISAM2,DISP=SHR,UNIT=&UISAM,VOLUME=&VISAM2 00003000
//DLIB DD DSN=&LIB.L,DISP=&LIBDISP,VOLUME=&VLIB,UNIT=&ULIB 00003100
//TLIB DD DSN=&LIB1.L,DISP=SHR,VOLUME=&VLIB1,UNIT=&ULIB1 00003200
//SOURCPRD DD SYSOUT=(&A,&CL)                             00003300
//PRINTER DD SYSOUT=(&A,&CL)                              00003400
//SYSOUT DD SYSOUT=(&A,&CL)                               00003500
//SYSPRINT DD SYSOUT=(&A,&CL)                             00003600
//SYSUDUMP DD SYSOUT=(&A,&CL1)                             00003700
//OPLINE DD SYSOUT=(&A,&CL2),DCB=LRECL=133                00003800
//OPIWCF DD UNIT=&STG,SPACE=(TRK,(2,1)),DCB=(BLKSIZE=400,LRECL=80, *00003900
//          RECFM=FB),DSN=&CHKID.A,DISP=(, &NRMDSP,&CHKDSP) 00004000
//OPCOMREC DD UNIT=&STG,SPACE=(TRK,(1,1)),                X00004100
//          DISP=(, &NRMDSP,&CHKDSP),DSN=&CHKID.B          00004200
//OPLGOGO DD UNIT=&STG,SPACE=(TRK,(20,1)),                X00004300
//          DISP=(, &NRMDSP,&CHKDSP),DSN=&CHKID.C          00004400
//OPLEXEX DD UNIT=(&STG,SEP=OPLGOGO),SPACE=(TRK,(20,1)), X00004500
//          DISP=(, &NRMDSP,&CHKDSP),DSN=&CHKID.D          00004600
//OPLEXDIR DD UNIT=(&STG,SEP=(OPLGOGO,OPLEXEX)),SPACE=(TRK,(20,1)), X00004700
//          DISP=(, &NRMDSP,&CHKDSP),DSN=&CHKID.E          00004800
//OPPUNCH DD SYSOUT=&B                                     00004900
//SLIB DD DSN=&.DLIB,DISP=SHR,UNIT=&ULIB,VOLUME=&VLIB      00005000
// DD DSN=&.TLIB,DISP=SHR,UNIT=&ULIB1,VOLUME=&VLIB1        00005100
// DD DSN=&LIB2.L,DISP=SHR,UNIT=&ULIB2,VOLUME=&VLIB2        00005200
// DD DSN=&JOBLIB,DISP=SHR                                00005300
//QDFILE DD DSN=&QDF,UNIT=&UQDF,VOLUME=&VQDF,DISP=(SHR,&QDISP,KEEP) 00005400
//QRTFILE DD DSN=&QRT,UNIT=&UQRT,VOLUME=&VQRT,DISP=(SHR,&QDISP,KEEP) 00005500

```

## NIPS 360 FFS

## PROCEDURES

```
//CHECKDD DD DSN=&CHKID,DISP=(&CHKST,&NPMDSP,&CHKDSP),UNIT=&UCHK, 00005600
//          VOLUME=&VCHK,SPACE=(CYL,(&CHKSP,1)) 00005700
//STAT DD UNIT=&STG,SPACE=(CYL,(2,1)) 00005800
//TRANST DD DSN=&TRANS,VOL=&VTRANS,UNIT=&UTRANS, 00005900
//          DISP=&TDISP,SPACE=(TRK,1) 00006000
```



## NIPS 360 FFS

## PROCEDURES

```

//XOPSD  PROC  A=A,B,BSZFILE=,BSZFIL1=,BSZFIL2=, 00000100
//          CL=',',CL1=',',CL2=',', 00000200
//          ISAM='DUMMY.FILE',ISAM1='DUMMY.FILE',ISAM2='DUMMY.FILE', 00000300
//          JOBLIB='FFS.JOBLIB',JOBMAC='FFS.JOBMACRO', 00000400
//          LAB=SL, 00000500
//          LIB='DUMMY.FILE',LIB1='DUMMY.FILE',LIB2='DUMMY.FILE', 00000600
//          LIBDISP=SHR, 00000700
//          PTFJOBL='PTF.JOBLIB',PTFJOBM='PTF.JOBMACRO', 00000710
//          RGN=100K, 00000800
//          SAM='DUMMY.FILE',SAM1='DUMMY.FILE',SAM2='DUMMY.FILE', 00000900
//          SDISP=SHR,SOURCL='DUMMY.FILE',STG=NIPW, 00001000
//          TDISP=MOD,TRANS='&&TRANS', 00001100
//          UISAM='(2314,P)', 00001200
//          ULIB=2314,ULIB1=2314,ULIB2=2314, 00001300
//          USAM='(TAPE9,,DEFER)',USOURCL=2314, 00001400
//          UTRANS=NIPW, 00001500
//          VISAM=,VISAM1=,VISAM2=, 00001600
//          VLIB=,VLIB1=,VLIB2=, 00001700
//          VSAM=,VSAM1=,VSAM2=, 00001800
//          VSOURCL=,VTRANS= 00001900
//** 00002000
//** CHARLES W. HICKISCH MAJOR,USA PROJ CODE=763NIPS BRANCH=431 00002100
//** DATE=MARCH 1, 1974 00002200
//** 00002300
//OP EXEC PGM=OP,REGION=&RGN 00002400
//STEPLIB DD DSN=&PTFJOBL,DISP=SHR 00002500
//          DD DSN=&JOBLIB,DISP=SHR 00002510
//DATAFILE DD DSNAME=&ISAM,DISP=SHR,UNIT=&UISAM,VOLUME=&VISAM 00002600
//DATAFIL1 DD DSNAME=&ISAM1,DISP=SHR,UNIT=&UISAM,VOLUME=&VISAM1 00002700
//DATAFIL2 DD DSNAME=&ISAM2,DISP=SHR,UNIT=&UISAM,VOLUME=&VISAM2 00002800
//SAMFILE DD DSNAME=&SAM.S,DISP=SHR,UNIT=&USAM,VOLUME=&VSAM, 00002900
//          LABEL=(, &LAB),DCB=BLKSIZE=&BSZFILE 00003000
//SAMFILE1 DD DSNAME=&SAM1.S,DISP=SHR,UNIT=&USAM,VOLUME=&VSAM1, 00003100
//          LABEL=(, &LAB),DCB=BLKSIZE=&BSZFIL1 00003200
//SAMFILE2 DD DSNAME=&SAM2.S,DISP=SHR,UNIT=&USAM,VOLUME=&VSAM2, 00003300
//          LABEL=(, &LAB),DCB=BLKSIZE=&BSZFIL2 00003400
//DLIB DD DSNAME=&LIB.L,DISP=&LIBDISP,VOLUME=&VLIB,UNIT=&ULIB 00003500
//TLIB DD DSNAME=&LIB1.L,DISP=SHR,VOLUME=&VLIB1,UNIT=&ULIB1 00003600
//SOURCPRT DD SYSOUT=(&A,&CL) 00003700
//SOURCLIB DD DISP=&SDISP,DSN=&SOURCL.L,VOL=&VSOURCL,UNIT=&USOURCL 00003800
//SYSPRINT DD SYSOUT=(&A,&CL),DCB=(LRECL=133,BLKSIZE=665) 00003900
//SYSOUT DD SYSOUT=(&A,&CL) 00004000
//SYSUDUMP DD SYSOUT=(&A,&CL1) 00004100
//SYSUT1 DD UNIT=&STG,SPACE=(TRK,(200,10)) 00004200
//SYSUT2 DD UNIT=(&STG,SEP=SYSUT1),SPACE=(TRK,(200,20)) 00004300
//SYSUT3 DD UNIT=(&STG,SEP=(SYSUT1, SYSUT2)),SPACE=(TRK,(200,20)) 00004400
//SYSPUNCH DD UNIT=&STG,SPACE=(TRK,(50,10)), 00004500
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=400,BUFNO=5) 00004600
//SYSLIB DD DSN=&PTFJOBM,DISP=SHR 00004700
//          DD DSN=&JOBMACRO,DISP=SHR 00004710
//SYSLMOD DD UNIT=&STG,SPACE=(TRK,(50,25,5)),DCB=SYSL.LINKLIB, 00004800
//          LABEL=RETPD=0000 00004900
//SYSTPRT DD SYSOUT=(&A,&CL),DCB=(RECFM=FB,LRECL=121,BLKSIZE=605) 00005000
//OPIWCF DD UNIT=&STG,SPACE=(TRK,(2,1)),DCB=(BLKSIZE=400,LRECL=80, 00005100
//          RECFM=FB) 00005200
//OPSTST DD UNIT=&STG,SPACE=(TRK,(10,2)) 00005300
//OPCREATE DD UNIT=&STG,SPACE=(TRK,(1,1)) 00005400

```

## NIPS 360 FFS

## PROCEDURES

```

//OPCOMREC DD UNIT=&STG,SPACE=(TRK,(1,1)) 00005500
//INSTS DD UNIT=&STG,SPACE=(TRK,(50,5)) 00005600
//LCF DD UNIT=&STG,SPACE=(TRK,(10,5)) 00005700
//LCT DD UNIT=&STG,SPACE=(TRK,(10,5)) 00005800
//LITSTR DD UNIT=&STG,SPACE=(TRK,(50,5)) 00005900
//NAMDEF DD DSNAME=*.SYSUT1,VOLUME=REF=*.SYSUT1,DISP=(OLD,PASS) 00006000
//LITAB DD DSNAME=*.SYSUT2,VOLUME=REF=*.SYSUT2,DISP=(OLD,PASS) 00006100
//RGTAB DD DSNAME=*.SYSUT3,VOLUME=REF=*.SYSUT3,DISP=(OLD,PASS) 00006200
//CGC DD UNIT=&STG,SPACE=(TRK,(1,1)) 00006300
//ERPTAB DD UNIT=&STG,SPACE=(TRK,(8,5)) 00006400
//SUBTAB DD UNIT=&STG,SPACE=(TRK,(1,1)) 00006500
//DCTNY DD DSNAME=*.SYSPUNCH,VOLUME=REF=*.SYSPUNCH,DISP=(OLD,PASS) 00006600
//OPLGOGO DD DSNAME=*.SYSUT1,VOLUME=REF=*.SYSUT1,DISP=(OLD,PASS) 00006700
//OPLEXEX DD DSNAME=*.SYSUT2,VOLUME=REF=*.SYSUT2,DISP=(OLD,PASS) 00006800
//OPLEXDIR DD DSNAME=*.SYSUT3,VOLUME=REF=*.SYSUT3,DISP=(OLD,PASS) 00006900
//OPLINE DD SYSOUT=(&A,&CL2),DCB=LRECL=133 00007000
//OPPUNCH DD SYSOUT=&B 00007100
//SLIB DD DSNAME=*.SYSLMOD,VOLUME=REF=*.SYSLMOD,DISP=(OLD,PASS) 00007200
// DD DSNAME=*.DLIB,DISP=SHR,UNIT=&ULIB,VOLUME=&VLIB 00007300
// DD DSNAME=*.TLIB,DISP=SHR,UNIT=&ULIB1,VOLUME=&VLIB1 00007400
// DD DSNAME=&LIB2.L,DISP=SHR,UNIT=&ULIB2,VOLUME=&VLIB2 00007500
// DD DSNAME=&JOBLIB,DISP=SHR 00007600
//SYSAIN DD DSNAME=*.LITSTR,VOLUME=REF=*.LITSTR,DISP=(OLD,PASS), 00007700
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=400,BUFNO=5) 00007800
// DD DSNAME=*.INSTS,VOLUME=REF=*.INSTS,DISP=(OLD,PASS), 00007900
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=400,BUFNO=5) 00008000
//SYSLIN DD DSNAME=*.LCF,VOLUME=REF=*.LCF,DISP=(OLD,PASS) 00008100
//SYSTIN DD DSNAME=*.LCT,VOLUME=REF=*.LCT,DISP=(OLD,PASS) 00008200
//LOADR DD DSNAME=*.SYSPUNCH,VOLUME=REF=*.SYSPUNCH,DISP=(OLD,PASS), X00C08300
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=400) 00008400
//STAT DD DSN=*.INSTS,VOL=REF=*.INSTS,DISP=(OLD,PASS) 00008500
//TRANST DD DSN=&TRANS,VOL=&VTRANS,UNIT=&UTRANS, 00008600
// DISP=&TDISP,SPACE=(TRK,1) 00008700

```

## NIPS 360 FFS

## PROCEDURES

```

//X7PSDEX PROC A=A,B=B,BSZFILE=,BSZFIL1=,BSZFIL2=,
// CL=',' ,CL1=',' ,CL2=',' ,
// CHKDSP=DELETE,CHKID='&&CHKID',CHKSP=0,CHKST=NEW,
// ISAM='DUMMY.FILE',ISAM1='DUMMY.FILE',ISAM2='DUMMY.FILE',
// JOBLIB='FFS.JOBLIB',
// LAB=SL,
// LIB='DUMMY.FILE',LIB1='DUMMY.FILE',LIB2='DUMMY.FILE',
// LIBDISP=SHR,
// NRMCDSP=DELETE,
// PTFJOBL='PTF.JOBLIB',
// RGN=100K,
// SAM='DUMMY.FILE',SAM1='DUMMY.FILE',SAM2='DUMMY.FILE',
// STG=NIPW,
// TDISP=MOD,TRANS='&&TRANS',
// UCHK=NIPW,
// UISAM='(2314,P)',
// ULIB=2314,ULIB1=2314,ULIB2=2314,
// USAM='(TAPE9,,DEFER)',
// UTRANS=NIPW,
// VISAM=,VISAM1=,VISAM2=,
// VLIB=,VLIB1=,VLIB2=,
// VSAM=,VSAM1=,VSAM2=,
// VCHK=,VTRANS=
// **
// * CHARLES W. HICKTSCH MAJOR,USA PRJ CODE=763NIPS BRANCH=431
// * DATE=MARCH 1, 1974
// *
//OP EXEC PGM=OP,REGION=&RGN
//STEPLIB DD DSN=&PTFJOBL,DISP=SHR
// DD DSN=&JOBLIB,DISP=SHR
//DATAFILE DD DSN=&ISAM,DISP=SHR,UNIT=&UISAM,VOLUME=&VISAM
//DATAFIL1 DD DSN=&ISAM1,DISP=SHR,UNIT=&UISAM,VOLUME=&VISAM1
//DATAFIL2 DD DSN=&ISAM2,DISP=SHR,UNIT=&UISAM,VOLUME=&VISAM2
//SAMFILE DD DSN=&SAM.S,DISP=SHR,UNIT=&USAM,VOLUME=&VSAM,
// LABEL=(, &LAB),DCB=BLKSIZE=&BSZFILE
//SAMFILE1 DD DSN=&SAM1.S,DISP=SHR,UNIT=&USAM,VOLUME=&VSAM1,
// LABEL=(, &LAB),DCB=BLKSIZE=&BSZFIL1
//SAMFILE2 DD DSN=&SAM2.S,DISP=SHR,UNIT=&USAM,VOLUME=&VSAM2,
// LABEL=(, &LAB),DCB=BLKSIZE=&BSZFIL2
//DLIB DD DSN=&LIB.L,DISP=&LIBDISP,VOLUME=&VLIB,UNIT=&ULIB
//TLIB DD DSN=&LIB1.L,DISP=SHR,VOLUME=&VLIB1,UNIT=&ULIB1
//SOURCPRT DD SYSOUT=(&A,&CL)
//PRINTER DD SYSOUT=(&A,&CL)
//SYSOUT DD SYSOUT=(&A,&CL)
//SYSPRINT DD SYSOUT=(&A,&CL),DCB=LRECL=133
//SYSUDUMP DD SYSOUT=(&A,&CL1)
//OPLINE DD SYSOUT=(&A,&CL2),DCB=LRECL=133
//OPIWCF DD UNIT=&STG,SPACE=(TRK,(2,1)),DCB=(BLKSIZE=400,LRECL=80,
// RECFM=FB),DSNAME=&CHKID.A,DISP=(, &NRMDSP,&CHKDSP)
//OPCOMREC DD UNIT=&STG,SPACE=(TRK,(1,1)),
// DISP=(, &NRMDSP,&CHKDSP),DSNAME=&CHKID.B
//OPLGOGO DD UNIT=&STG,SPACE=(TRK,(20,1)),
// DISP=(, &NRMDSP,&CHKDSP),DSNAME=&CHKID.C
//OPLEXEX DD UNIT=( &STG,SEP=OPLGOGO),SPACE=(TRK,(20,1)),
// DISP=(, &NRMDSP,&CHKDSP),DSNAME=&CHKID.D
//OPLEXDIR DD UNIT=( &STG,SEP=(CPLGOGO,OPLEXEX)),SPACE=(TRK,(20,1)),
// DISP=(, &NRMDSP,&CHKDSP),DSNAME=&CHKID.E

```

## NIPS 360 FFS

## PROCEDURES

```
//OPPUNCH DD SYSOUT=&B 00005600
//SLIB DD DSN=&*.DLIB,DISP=SHR,UNIT=&ULIB,VOLUME=&VLIB 00005700
// DD DSN=&*.TLIB,DISP=SHR,UNIT=&ULIB1,VOLUME=&VLIB1 00005800
// DD DSN=&LIB2.L,DISP=SHR,UNIT=&LLIB2,VOLUME=&VLIB2 00005900
// DD DSN=&JOBLIB,DISP=SHR 00006000
//CHECKDD DD DSN=&CHKID,DISP=(&CHKST,&NRMDSP,&CHKDSP),UNIT=&UCHK, 00006100
// VOLUME=&VCHK,SPACE=(CYL,(&CHKSP,1)) 00006200
//STAT DD UNIT=&STG,SPACE=(CYL,(2,1)) 00006300
//TRANST DD DSN=&TRANS,VOL=&VTRANS,UNIT=&UTRANS, 00006400
// DISP=&TDISP,SPACE=(TRK,1) 00006500
```



## NIPS 360 FFS

## PROCEDURES

```

//XQRTQDF PROC A=A,BSZNEWF=1004,CL=',',CL1=',',DEN=,
//      JOBLIB='FFS.JOBLIB',
//      LAB=,QDF='&&QDFILE',QDISP=PASS,
//      PTFJOBL='PTF.JOBLIB',
//      QRT='&&QRTFILE',RGN=60K,SAM='DUMMY',TRCH=,
//      UQDF=NIPW,UQPT=NIPW,USAM='(TAPE9,,DEFER)',
//      VQDF=,VQRT=,VSAM=
//*
//*   CHARLES W. HICKISCH   MAJOR,USA   PROJ CODE=763NIPS   BRANCH=431
//*   DATE=MARCH 1, 1974
//*
//QRTQDF EXEC PGM=UTQRTQDF,REGION=&RGN
//STEPLIB DD DSN=&PTFJOBL,DISP=SHR
//      DD DSN=&JOBLIB,DISP=SHR
//SYSPRINT DD SYSOUT=(&A,&CL)
//SYSUDUMP DD SYSOUT=(&A,&CL1)
//QRTFILE DD DSNAME=&QRT,UNIT=&UQRT,VOLUME=&VQRT,DISP=(SHR,&QDISP,KEEP)
//QDFILE DD DSNAME=&QDF,UNIT=&UQDF,VOLUME=&VQDF,DISP=(SHR,&QDISP,KEEP)
//SAMFILE DD DSNAME=&SAM.S,UNIT=&USAM,VOLUME=&VSAM,DISP=(,KEEP),
//      DCB=(TRTCH=&TRCH,DEN=&DEN,BLKSIZE=&BSZNEWF),
//      LABEL=(,&LAB)

```

```

00000100
00000200
00000300
00000310
00000400
00000500
00000600
00000700
00000800
00000900
00001000
00001100
00001200
00001210
00001300
00001400
00001500
00001600
*00001700
00001800
00001900

```

## NIPS 360 FFS

## PROCEDURES

```

//XQUIP  PROC  A=A,                                00000100
//          CL=',',CL1=',',CL2=',',                00000200
//          ISAM1='DUMMY.FILE',ISAM2='DUMMY.FILE',  00000300
//          JOBLIB='FFS.JOBLIB',                   00000400
//          LIB='DUMMY.FILE',LIB1='DUMMY.FILE',     00000500
//          PTFJOBL='PTF.JOBLIB',                  00000510
//          QRT='&&QRTFILE',                        00000600
//          QDF='&&QDFILE',QDISP=PASS,              00000700
//          RGN=92K,SDISP=SHR,                      00000800
//          SOURCL='DUMMY.FILE',                   00000900
//          STG=NIPW,TDISP=MOD,TRANS='&&TRANS',     00001000
//          UISAM1='(2314,P)',UISAM2='(2314,P)',    00001100
//          ULIB=2314,ULIB1=2314,                  00001200
//          UQDF=NIPW,                             00001300
//          UQRT=NIPW,                             00001400
//          USOURCL=2314,UTRANS=NIPW,              00001500
//          VISAM1=,VISAM2=,                       00001600
//          VLIB=,VLIB1=,                          00001700
//          VQDF=,                                  00001800
//          VQRT=,                                  00001900
//          VSOURCL=,VTRANS=                       00002000
//**                                                00002100
//**  CHARLES W. HICKISCH  MAJOR,USA  PR CJ CODE=763NIPS BRANCH=431  00002200
//**  DATE=MARCH 1, 1974                                         00002300
//**                                                           00002400
//QUIP EXEC PGM=IPBASE,REGION=&RGN                    00002500
//STEPLIB DD DSN=&PTFJOBL,DISP=SHR                    00002600
//          DD DSN=&JOBLIB,DISP=SHR                  00002610
//SYSUT1 DD SPACE=(TRK,(50,10)),UNIT=&STG             00002700
//SYSUT2 DD SPACE=(TRK,(1,1)),UNIT=&STG              00002800
//SYSUT3 DD SPACE=(TRK,(10,10)),UNIT=(&STG,SEP=SYSUT1) 00002900
//SLIB DD DSN=&LIB.L,UNIT=&ULIB,VOLUME=&VLIB,DISP=SHR  00003000
//          DD DSN=&LIB1.L,UNIT=&ULIB1,VOLUME=&VLIB1,DISP=SHR 00003100
//          DD DSN=&JOBLIB,DISP=SHR                  00003200
//QDFILE DD DSN=&QDF,UNIT=&UQDF,VOLUME=&VQDF,DISP=(SHR,&QDISP,KEEP) 00003300
//QRTFILE DD DSN=&QRT,UNIT=&UQRT,VOLUME=&VQRT,DISP=(SHR,&QDISP,KEEP) 00003400
//DATAFIL1 DD DSN=&ISAM1,UNIT=&UISAM1,VOL=&VISAM1,DISP=SHR 00003500
//DATAFIL2 DD DSN=&ISAM2,UNIT=&UISAM2,VOL=&VISAM2,DISP=SHR 00003600
//SOURCLIB DD DISP=&SDISP,DSN=&SOURCL.L,VOL=&VSOURCL,UNIT=&USOURCL 00003700
//SOURCPRT DD SYSOUT=(&A,&CL)                      00003800
//QUERYQUE DD SPACE=(TRK,(1,1)),UNIT=&STG            00003900
//QUERYLOG DD SYSOUT=(&A,&CL),SPACE=(TRK,(1,1))     00004000
//SYSUDUMP DD SYSOUT=(&A,&CL1)                      00004100
//SYSPRINT DD SYSOUT=(&A,&CL2)                      00004200
//STAT DD DSN=*.SYSUT1,VOL=REF=*.SYSUT1,DISP=(OLD,PASS) 00004300
//TRANST DD DSN=&TRANS,VOL=&VTRANS,UNIT=&UTRANS,DISP=&TDISP, 00004400
//          SPACE=(TRK,1)                             00004500

```

## NIPS 360 FFS

## PROCEDURES

```

//XQUIPSD PROC A=A,BSZFILE=,                                00000100
//      CL=' ',CL1=' ',CL2=' ',                                00000200
//      ISAM='DUMMY.FILE',                                    00000300
//      ISAM1='DUMMY.FILE',ISAM2='DUMMY.FILE',                00000400
//      JOBLIB='FFS.JOBLIB',                                  00000500
//      LAB=SL,                                                00000600
//      LIB='DUMMY.FILE',LIB1='DUMMY.FILE',                    00000700
//      PTFJOBL='PTF.JOBLIB',                                  00000710
//      QDFSP='(5,1)',                                         00000800
//      QRTSP='(2,1)',                                         00000900
//      RGN=92K,                                                00001000
//      SAM='DUMMY.FILE',SDISP=SHR,                             00001100
//      SORTSP='(2,1)',                                         00001200
//      SOURCL='DUMMY.FILE',                                    00001300
//      STG=NIPW,TDISP=MOD,TRANS='&&TRANS',                    00001400
//      UISAM='(2314,P)',                                       00001500
//      UISAM1='(2314,P)',UISAM2='(2314,P)',                  00001600
//      ULIB=2314,ULIB1=2314,                                   00001700
//      USAM='(TAPE9,,DEFER)',                                  00001800
//      USOURCL=2314,UTRANS=NIPW,                               00001900
//      VISAM=,                                                 00002000
//      VISAM1=,VISAM2=,                                       00002100
//      VLIB=,VLIB1=,                                          00002200
//      VSAM=,                                                  00002300
//      VSOURCL=,VTRANS=,                                       00002400
//      XINDEX='DUMMY.FILE',                                   00002500
//      XUNIT=2314,XVOL=                                       00002600
//**                                                         00002700
//** CHARLES W. HICKISCH MAJOR,USA PROJ CODE=763NIPS BRANCH=431 00002800
//** DATE=MARCH 1, A974                                       00002900
//**                                                         00003000
//QUIP EXEC PGM=IPBASE,REGION=&RGN                             00003100
//STEPLIB DD DSN=&PTFJOBL,DISP=SHR                             00003200
//      DD DSN=&JOBLIB,DISP=SHR                                00003210
//SYSUT1 DD SPACE=(TRK,(50,10)),UNIT=&STG                      00003300
//SYSUT2 DD SPACE=(TRK,(1,1)),UNIT=&STG                        00003400
//SYSUT3 DD SPACE=(TRK,(10,10)),UNIT=(&STG,SEP=SYSUT1)        00003500
//SYSUT4 DD SPACE=(CYL,(0,0)),UNIT=&STG                        00003600
//SLIB DD DSNAME=&LIB.L,UNIT=&ULIB,VOLUME=&VLIB,DISP=SHR       00003700
//      DD DSNAME=&LIB1.L,UNIT=&ULIB1,VOLUME=&VLIB1,DISP=SHR   00003800
//      DD DSNAME=&JOBLIB,DISP=SHR                             00003900
//DATAFILE DD DISP=SHR,DSNAME=&ISAM,UNIT=&UISAM,VOLUME=&VISAM   00004000
//DATAFIL1 DD DSN=&ISAM1,UNIT=&UISAM1,VOL=&VISAM1,DISP=SHR     00004100
//DATAFIL2 DD DSN=&ISAM2,UNIT=&UISAM2,VOL=&VISAM2,DISP=SHR     00004200
//SAMFILE DD DISP=SHR,DSNAME=&SAM.S,UNIT=&USAM,VOLUME=&VSAM,    *00004300
//      LABEL=(, &LAB),CCB=(RECFM=VB,LRECL=1000,BLKSIZE=&BSZFILE) 00004400
//QUERYLOG DD SYSOUT=(&A,&CL),SPACE=(TRK,(1,1))               00004500
//SOURCLIB DD DISP=&SDISP,DSN=&SOURCL.L,VOL=&VSOURCL,UNIT=&USOURCL 00004600
//SOURCPT DD SYSOUT=(&A,&CL)                                    00004700
//SYSUDUMP DD SYSOUT=(&A,&CL1)                                  00004800
//SYSPRINT DD SYSOUT=(&A,&CL2)                                  00004900
//QUERYQUE DD SPACE=(TRK,(1,1)),UNIT=&STG                      00005000
//PB DD UNIT=&STG,SPACE=(CYL,&QDFSP)                             00005100
//KEY DD UNIT=&STG,SPACE=(CYL,&CRTSP)                           00005200
//SORTWK01 DD UNIT=&STG,SPACE=(CYL,&SORTSP)                     00005300
//SORTWK02 DD UNIT=&STG,SPACE=(CYL,&SORTSP)                     00005400
//SORTWK03 DD UNIT=&STG,SPACE=(CYL,&SORTSP)                     00005500

```

## NIPS 360 FFS

## PROCEDURES

//SORTWK04 DD UNIT=&STG,SPACE=(CYL,&SCRTSP)	00005600
//STAT DD DSN=*.SYSUT1,VOL=REF=*.SYSUT1,DISP=(OLD,PASS)	00005700
//TRANST DD DSN=&TRANS,VOL=&VTRANS,UNIT=&UTRANS,DISP=&TDISP,	00005800
//SPACE=(TRK,1)	00005900
//XINDEX DD DSN=&XINDEX.X,UNIT=&XUNIT,VCL=&XVCL,DISP=SHR	00006000



```

//XRASP PROC A=A,BSZFILE=,BSZFIL1=,BSZFIL2=, 00000100
// CL=',' ,CL1=',' ,ISAM='DUMMY.FILE',ISAM1='DUMMY.FILE', 00000200
// ISAM2='DUMMY.FILE',JOBLIB='FFS.JOBLIB', 00000300
// JOBMAC='FFS.JOBMACRO',LAB=SL,LIB='DUMMY.FILE', 00000400
// LIBDISP=SHR,LIB1='DUMMY.FILE',LIB2='DUMMY.FILE', 00000500
// PTFJOBL='PTF.JOBLIB',PTFJOBM='PTF.JOBMACRO', 00000510
// QDF='&&QDFILE',QDFSP='(50,10)',QDISP=PASS,QRT='&&QRTFILE', 00000600
// QRTSP='(10,10)',RGN=100K,SAM='DUMMY.FILE', 00000700
// SAM1='DUMMY.FILE',SAM2='DUMMY.FILE',SDISP=SHR, SORTSP=20, 00000800
// SOURCL='DUMMY.FILE',STG=NIPW,TDISP=MOD,TRANS='&&TRANS', 00000900
// UISAM='(2314,P)',ULIB=2314,ULIB1=2314,ULIB2=2314, 00001000
// UQDF=NIPW,UQRT=NIPW,USAM='(TAPE9,DEFER)',USOURCL=2314, 00001100
// UTRANS=NIPW,VISAM=,VISAM1=,VISAM2=,VLIB=,VLIB1=,VLIB2=, 00001200
// VQDF=,VQRT=,VSAM=,VSAM1=,VSAM2=,VSOURCL=,VTRANS=, 00001300
// XDISP=SHR,XINDEX='DUMMY.FILE',XINDEX1='DUMMY.FILE', 00001400
// XINDEX2='DUMMY.FILE',XUNIT=2314,XVOL=,XVOL1=,XVOL2= 00001500
// * 00001600
// * CHARLES W. HICKISCH MAJOR,USA PRGJ CODE=763NIPS BRANCH=431 00001700
// * DATE=MARCH 1, 1974 00001800
// * 00001900
//RASP EXEC PGM=RSEEXEC,REGION=&RGN 00002000
//STEPLIB DD DISP=SHR,DSN=&PTFJOBL 00002100
// DD DISP=SHR,DSN=&JOBLIB 00002110
//DATAFILE DD DSN=&ISAM,DISP=SHR,UNIT=&UISAM,VOLUME=&VISAM 00002200
//DATAFIL1 DD DSN=&ISAM1,DISP=SHR,UNIT=&UISAM,VOLUME=&VISAM1 00002300
//DATAFIL2 DD DSN=&ISAM2,DISP=SHR,UNIT=&UISAM,VOLUME=&VISAM2 00002400
//SAMFILE DD DSN=&SAM.S,DISP=SHR,UNIT=&USAM,VOLUME=&VSAM, 00002500
// LABEL=(,&LAB),DCB=BLKSIZE=&BSZFILE 00002600
//SAMFILE1 DD DSN=&SAM1.S,DISP=SHR,UNIT=&USAM,VOLUME=&VSAM1, 00002700
// LABEL=(,&LAB),DCB=BLKSIZE=&BSZFIL1 00002800
//SAMFILE2 DD DSN=&SAM2.S,DISP=SHR,UNIT=&USAM,VOLUME=&VSAM2, 00002900
// LABEL=(,&LAB),DCB=BLKSIZE=&BSZFIL2 00003000
//QDFILE DD DSN=&QDF,UNIT=(&UQDF,SEP=DATAFILE),VOLUME=&VQDF, 00003100
// SPACE=(CYL,&QDFSP),DISP=(,&QDISP,DELETE) 00003200
//SORTOUT DD DSN=&QRT,UNIT=(&UQRT,SEP=DATAFILE,QDFILE)), 00003300
// VOLUME=&VQRT,SPACE=(CYL,&QRTSP),DISP=(,&QDISP,DELETE), 00003400
// DCB=(RECFM=VB,LRECL=1011,BLKSIZE=1015) 00003500
//SORTIN DD DSN=&*.SORTOUT,VOLUME=REF=&*.SORTOUT,DISP=(OLD,PASS) 00003600
//SORTWK01 DD SPACE=(CYL,(&SORTSP),,CONTIG),UNIT=(&STG,SEP=SORTIN) 00003700
//SORTWK02 DD SPACE=(CYL,(&SORTSP),,CONTIG), 00003800
// UNIT=(&STG,SEP=SORTWK01),SEP=SCRTWK01 00003900
//SORTWK03 DD SPACE=(CYL,(&SORTSP),,CONTIG), 00004000
// UNIT=(&STG,SEP=SORTWK02),SEP=SORTWK02 00004100
//SORTWK04 DD SPACE=(CYL,(&SORTSP),,CONTIG), 00004200
// UNIT=(&STG,SEP=(SORTWK01,SORTWK03)), 00004300
// SEP=(SORTWK01,SORTWK03) 00004400
//SORTWK05 DD SPACE=(CYL,(&SORTSP),,CONTIG), 00004500
// UNIT=(&STG,SEP=(SORTWK02,SORTWK04)), 00004600
// SEP=(SORTWK02,SORTWK04) 00004700
//SORTWK06 DD SPACE=(CYL,(&SORTSP),,CONTIG), 00004800
// UNIT=(&STG,SEP=(SORTWK01,SORTWK03,SORTWK05)), 00004900
// SEP=(SORTWK01,SORTWK03,SORTWK05) 00005000
//SYSLMOD DD UNIT=&STG,SPACE=(TRK,(40,20,20)),DCB=SYSL.LINKLIB, 00005100
// LABEL=EXPDT=66366 00005200
//DLIB DD DSN=&LIB.L,DISP=&LIBDISP,VOLUME=&VLIB,UNIT=&ULIB 00005300
//TLIB DD DSN=&LIB1.L,DISP=SHR,VOLUME=&VLIB1,UNIT=&ULIB1 00005400
//SLIB DD DSN=&*.SYSLMOD,DISP=(OLD,PASS),VOLUME=REF=&*.SYSLMOD 00005500

```

## NIPS 360 FFS

## PROCEDURES

```

// DD DSNAME=*.DLIB,DISP=SHR,UNIT=&ULIB,VOLUME=&VLIB          00005600
// DD DSNAME=*.TLIB,DISP=SHR,UNIT=&ULIB1,VOLUME=&VLIB1         00005700
// DD DSNAME=&LIB2.L,DISP=SHR,UNIT=&ULIB2,VOLUME=&VLIB2        00005800
// DD DSNAME=&JOBLIB,DISP=SHR                                  00005900
//SORTLIB DD DISP=SHR,DSNAME=SYS1.SORTLIB                      00006000
//SYSLIB DD DISP=SHR,DSN=&PTFJOBM                              00006100
// DD DISP=SHR,DSN=&JOBMAC                                     00006110
// DD DISP=SHR,DSNAME=SYS1.MACLIB                             00006200
//DPUNCH DD DSNAME=*.SORTWK04,VOLUME=REF=*.SORTWK04,DISP=(OLD,PASS) 00006300
//SYSUT1 DD DSNAME=*.SORTWK01,VOLUME=REF=*.SORTWK01,DISP=(OLD,PASS) 00006400
//SYSUT2 DD DSNAME=*.SORTWK05,VOLUME=REF=*.SORTWK05,DISP=(OLD,PASS) 00006500
//STEREO DD DSNAME=*.SORTWK06,VOLUME=REF=*.SORTWK06,DISP=(OLD,PASS) 00006600
//STROUTDD DD UNIT=&STG,SPACE=(TRK,(50,20))                  00006700
//SYSUT3 DD DSNAME=*.SORTWK06,VOLUME=REF=*.SORTWK06,DISP=(OLD,PASS), 00006800
// DCB=(BLKSIZE=80,LRECL=80)                                00006900
//ASSEMBIN DD UNIT=&STG,SPACE=(TRK,(50,5)),DCB=BLKSIZE=400,    C00007000
// DISP=(MOD,DELETE)                                         00007100
//RSIPDS DD UNIT=&STG,SPACE=(TRK,(3,1,1))                    00007200
//NAMETAB DD DSNAME=*.ASSEMBIN,VOLUME=REF=*.ASSEMBIN,DISP=(OLD,PASS) 00007300
//SOURCPRT DD SYSOUT=(&A,&CL)                                00007400
//SOURCLIB DD DISP=&SDISP,DSN=&SOURCL.L,VOL=&VSCURCL,UNIT=&USOURCL 00007500
//SYSPRINT DD SYSOUT=(&A,&CL)                                00007600
//SYSOUT DD SYSOUT=(&A,&CL)                                  00007700
//SYSUDUMP DD SYSOUT=(&A,&CL1)                                00007800
//STAT DD DSN=*.ASSEMBIN,VOL=REF=*.ASSEMBIN,DISP=(OLD,PASS) 00007900
//TRANST DD DSN=&TRANS,VOL=&VTRANS,UNIT=&UTRANS,DISP=&TDISP,    00008000
// SPACE=(TRK,(1))                                           00008100
//XINDEX DD DSN=&XINDEX.X,VOL=&XVOL,UNIT=&XUNIT,DISP=&XDISP     00008200
//XINDEX1 DD DSN=&XINDEX1.X,VOL=&XVOL1,UNIT=&XUNIT,DISP=&XDISP 00008300
//XINDEX2 DD DSN=&XINDEX2.X,VOL=&XVOL2,UNIT=&XUNIT,DISP=&XDISP 00008400
//PRINTER DD SYSOUT=(&A,&CL)                                00008500
//SYSDUMMY DD DUMMY                                          00008600

```

## NIPS 360 FFS

## PROCEDURES

```

//XSP          PROC  A=A,BLKSIZE=560,BSZFILE=,BSZNEWF=,          00000100
//              CL=',' ,CL1=',' ,DEN=,ISAM='DUMMY.FILE',          00000200
//              JCBLIB='FFS.JOBLIB',LAB=SL,LIB='DUMMY.FILE',      00000300
//              NBRBLK=200,RGN=100K,SAM='DUMMY.FILE',             00000400
//              PTFJOBL='PTF.JOBLIB',                             00000410
//              SAMOUT='DUMMY',SORTSP=10,STG=NIPW,TRCH=,          00000500
//              UISAM=2314,ULIB=2314,USAM='(TAPE9,,DEFER)',        00000600
//              VISAM=,VLIB=,VSAM=,VSMOUT=,XDISP=CLD,             00000700
//              XINDEX='DUMMY.FILE',XUNIT=2314,XVCL=              00000800
//*****                                                00000900
//***                                                    ** 00001000
//***  THIS PROC IS USED TO GENERATE OR UPDATE A DISK RESIDENT  ** 00001100
//***  INDEX DATA SET BASED ON THE DISK RESIDENT ISAM DATA FILE. ** 00001200
//***                                                    ** 00001300
//***  //STEPNAME      EXEC  XSP,ISAM=AAAAAAA,VISAM='SER=XXXXXX', ** 00001400
//***                  XVOL='SER=YYYYYY',XDISP=(BBB),NBRBLK=DDDD ** 00001500
//***                                                    ** 00001600
//**  CHARLES W. HICKISCH MAJOR,USA  PRJ CODE=763NIPS  BRANCH=431 00001700
//**  DATE=MARCH 1, 1974                                     00001800
//***                                                    ** 00001900
//*****                                                00002000
//**                                                    ** 00002100
//UTXSP      EXEC  PGM=UTNDXSPC,REGION=&RGN                      00002200
//STEPLIB    DD  DISP=SHR,DSN=&PTFJOBL                          00002300
//           DD  DISP=SHR,DSN=&JOBLIB                          00002310
//INDEXPRT   DD  SYSOUT=(&A,&CL)                                00002400
//ISAMWORK   DD  UNIT=&STG,SPACE=(CYL,(10)),DCB=(DSORG=IS,RECFM=VB, 00002500
//           LRECL=1000,BLKSIZE=1004,OPTCD=IYLR,CYLOFL=2)        00002600
//NEWFILE    DD  DSN=&ISAM,DISP=SHR,VOL=&VISAM,UNIT=&UISAM        00002700
//SAMFILE     DD  DSN=&SAM.S,UNIT=&USAM,VOLUME=&VSAM,              00002800
//           DISP=(SHR,KEEP),LABEL=(,&LAB),DCB=(RECFM=VB,LRECL=1000, 00002900
//           BLKSIZE=&BSZFILE,TRTCH=&TRCH,DEN=&DEN)              00003000
//SLIB       DD  DSN=&LIB.L,VOLUME=&VLIB,UNIT=&ULIB,DISP=SHR      00003100
//           DD  DSN=FFS.JOBLIB,DISP=SHR                      00003200
//SORTLIB    DD  DSN=SYS1.SORTLIB,DISP=SHR                     00003300
//SORTWK01   DD  UNIT=&STG,SPACE=(CYL,(&SORTSP),,CONTIG)        00003400
//SORTWK02   DD  UNIT=&STG,SPACE=(CYL,(&SORTSP),,CONTIG)        00003500
//SORTWK03   DD  UNIT=&STG,SPACE=(CYL,(&SORTSP),,CONTIG)        00003600
//SORTWK04   DD  UNIT=&STG,SPACE=(CYL,(&SORTSP),,CONTIG)        00003700
//SORTWK05   DD  UNIT=&STG,SPACE=(CYL,(&SORTSP),,CONTIG)        00003800
//SORTWK06   DD  UNIT=&STG,SPACE=(CYL,(&SORTSP),,CONTIG)        00003900
//SOURCPRT   DD  SYSOUT=(&A,&CL)                                00004000
//SYSOUT     DD  SYSOUT=(&A,&CL1)                                00004100
//SYSPRINT   DD  SYSOUT=(&A,&CL)                                00004200
//SYSUDUMP   DD  SYSOUT=(&A,&CL1)                                00004300
//UTSAMOUT   DD  &SAMOUT.DCB=(RECFM=VB,LRECL=1000,BLKSIZE=&BSZNEWF, 00004400
//           DEN=&DEN,TRTCH=&TRCH),DISP=(,KEEP),LABEL=(,&LAB),    00004500
//           UNIT=&USAM,VOLUME=(PRIVATE,RETAIN,&VSMOUT),DSN=&SAM.S 00004600
//XINDEX     DD  DSN=&XINDEX.X,DISP=(&XDISP,KEEP),UNIT=&XUNIT,    00004700
//           VOL=&XVOL,SPACE=(&BLKSIZE,&NBRBLK),                00004800
//           DCB=(BLKSIZE=&BLKSIZE,RECFM=F,KEYLEN=4,DSORG=DA)    00004900

```

## NIPS 360 FFS

## PROCEDURES

```

//XSTOIS PROC  A=A,BSZFILE=,BSZNEWF=,CC=,CL=',' ,CL1=',' ,CYLOFL=1,DEN=, 00000100
//              INDEX=1,ISAM='DUMMY.FILE',JOBLIB='FFS.JCBLIB', 00000200
//              LAB=SL,NDISP=KEEP,OVFLOW=5,PRIME=40,RGN=60K, 00000300
//              PTFJOB=PTF.JOBLIB', 00000310
//              SAM='DUMMY.FILE',SEQNO=1,TRCH=,UISAM='(2314,P)', 00000400
//              USAM='(TAPE9,,DEFER)',VISAM='SER=CANCEL', 00000500
//              VCVFLOW='REF=*.DATAFILE',VPRIME='REF=*.DATAFILE', 00000600
//              VSAM= 00000700
//** 00000800
//** CHARLES W. HICKISCH MAJOR,USA PROJ CODE=763NIPS BRANCH=431 00000900
//** DATE=MARCH 1, 1974 00001000
//** 00001100
//STP1 EXEC PGM=UTBLDISM,REGION=&RGN,PARM='&CC' 00001200
//STEPLIB DD DISP=SHR,DSN=&PTFJOB 00001300
//          DD DISP=SHR,DSN=&JOBLIB 00001310
//DATAFILE DD DCB=(DSORG=IS,CYLOFL=&CYLOFL,BUFNO=5,BLKSIZE=&BSZNEWF), 00001400
//          VOLUME=&VISAM, 00001500
//          DSNNAME=&ISAM.(INDEX),SPACE=(CYL,&INDEX),UNIT=&UISAM, 00001600
//          DISP=(, &NDISP,DELETE) 00001700
//          DD DCB=*.DATAFILE,VOLUME=&VPRIME, 00001800
//          DSNNAME=&ISAM.(PRIME),SPACE=(CYL,&PRIME),UNIT=&UISAM, 00001900
//          DISP=(, &NDISP,DELETE) 00002000
//          DD DCB=*.DATAFILE,VOLUME=&VOVFLOW, 00002100
//          DSNNAME=&ISAM.(OVFLOW),SPACE=(CYL,&CVFLOW),UNIT=&UISAM, 00002200
//          DISP=(, &NDISP,DELETE) 00002300
//SAMFILE DD DSNNAME=&SAM.S,DISP=SHR,UNIT=&USAM,VOLUME=&VSAM, 00002400
//          LABEL=(&SEQNO,&LAB),DCB=(RECFM=VB,LRECL=1000,BLKSIZE=&BSZFILE, 00002500
//          DEN=&DEN,TRCH=&TRCH,BUFNO=5) 00002600
//SYSPRINT DD SYSOUT=(&A,&CL) 00002700
//SYSUDUMP DD SYSOUT=(&A,&CL1) 00002800

```



## NIPS 360 FFS

## PROCEDURES

//XSUBCHK	PROC A=A,CL=',',CL1=',',CL2=',',JOB LIB='FFS.JOB LIB',	00000100
//	PTFJOB L='PTF.JOB LIB',	00000110
//	LIB=,ULIB=2314,VLIB=	00000200
/**		00000300
/**	CHARLES W. HICKISCH MAJOR,USA PR CJ CODE=763NIPS BRANCH=431	00000400
/**		00000600
//SUBCHK	EXEC PGM=UTSUBCHK	00000700
//STEPLIB	DD DSN=&PTFJOB L,DISP=SHR	00000800
//	DD DISP=SHR,DSN=&JOB LIB	00000810
//SLIB	DD DSN=&LIB.L,VOL=&VLIB,UNIT=&ULIB,DISP=SHR	00000900
//SYSDMY	DD SYSOUT=(&A,&CL)	00001000
//SYSUDUMP	DD SYSOUT=(&A,&CL1)	00001100
//SYSPRINT	DD SYSOUT=(&A,&CL2)	00001200

## NIPS 360 FFS

## PROCEDURES

```

//XSUBLDR      PROC  A=A,CL=*,*,CL1=*,*,JOB LIB='FFS.JOB LIB',      00000100
//              JOB MAC='FFS.JOB MACRO',LIB=NONE,LIBDISP=OLD,        00000200
//              LIBSP='(2,1,5)',MODLIB='&TEMP',RGN=60K,             00000300
//              PTFJOB LIB='PTF.JOB LIB',PTFJOB M='PTF.JOB MACRO',   00000310
//              STG=NIPW,ULIB=2314,VLIB=,BLK=7294                   00000400
//**                                                    00000500
//*   CHARLES W. HICKISCH  MAJOR,USA  PROJ CODE=763NIPS  BRANCH=431  00000600
//*   DATE=MARCH 1, 1974                                           00000700
//**                                                    00000800
//SUB          EXEC  PGM=UTSUBLDR,REGION=&RGN                       00000900
//STEPLIB      DD   DISP=SHR,DSN=&PTFJOB L                          00001000
//              DD   DISP=SHR,DSN=&JOB LIB                          00001010
//SYSUDUMP      DD  SYSOUT=(&A,&CL1)                                00001100
//SYSPRINT      DD  SYSOUT=(&A,&CL)                                00001200
//ASSEMBIN      DD   UNIT=&STG,SPACE=(CYL,01),DSNAME=&AIN,DISP=(,PASS), *00001300
//              DCB=(BLKSIZE=400)                                00001400
//LNKEDIN       DD   UNIT=&STG,SPACE=(CYL,01),DSNAME=&CLIN,DISP=(,PASS), *00001500
//              DCB=(BLKSIZE=400)                                00001600
//SYSOUT        DD  SYSOUT=(&A,&CL)                                00001700
//ASMBL         EXEC  PGM=IEUASM,COND=(0,LT,SUB),REGION=&RGN        00001800
//SYSUT1        DD  SPACE=(1700,(400,50)),UNIT=&STG                00001900
//SYSUT2        DD  SPACE=(1700,(400,50)),UNIT=(&STG,SEP=SYSUT1)   00002000
//SYSUT3        DD  SPACE=(1700,(400,50)),UNIT=(&STG,SEP=(SYSUT1,SYSUT2)) 00002100
//SYSLIB        DD   DISP=SHR,DSN=&PTFJOB M                        00002200
//              DD   DISP=SHR,DSN=&JOB MACRO                      00002210
//              DD   DISP=SHR,DSN=SYS1.MACLIB                     00002300
//SYSPRINT      DD  SYSOUT=(&A,&CL)                                00002400
//SYSPUNCH      DD   UNIT=&STG,SPACE=(CYL,5),DSNAME=&LKIN,DISP=(,PASS), *00002500
//              DCB=(BLKSIZE=400)                                00002600
//SYSIN         DD   DISP=(OLD,DELETE),DSNAME=&AIN                00002700
//LKEDT         EXEC  PGM=IEWL,PARM='LIST,XREF',REGION=&RGN,        X00002800
//              COND=((0,LT,ASMBL),(0,LT,SUB))                    00002900
//SYSUT1        DD   SPACE=(TRK,(50,5)),UNIT=&STG                 00003000
//SYSPRINT      DD  SYSOUT=(&A,&CL)                                00003100
//SYSLMOD       DD   DSNAME=&LIB.L,UNIT=&ULIB,VOLUME=&VLIB,        X00003200
//              SPACE=(CYL,&LIBSP),DISP=(&LIBDISP,KEEP),          X00003300
//              DCB=(RECFM=U,BLKSIZE=&BLK)                        00003400
//DPPUNCH       DD   DSNAME=&LKIN,DISP=(OLD,DELETE)               00003500
//SYSLIN        DD   DSNAME=&CLIN,DISP=(OLD,DELETE)               00003600
//SYSLIB        DD   DISP=SHR,DSNAME=&JOB LIB                     00003700
//MODLIB        DD   DSNAME=&MODLIB,DISP=(OLD,DELETE)             00003800

```

## NIPS 360 FFS

## PROCEDURES

```

//XTABGEN      PROC  A=A,BLK=7294,CL=',',CL1=',',CL2=',',
//              JOBLIB='FFS.JOBLIB',LIB=NONE,LIBDISP=OLD,
//              LIBSP='(2,1,5)',RGN=80K,SCRTSP=8,STG=NIPW,ULIB=2314,
//              PTFJOBL='PTF.JOBLIB',
//              VLIB=
//**
//**  CHARLES W. HICKISCH  MAJOR,USA  PRJ CODE=763NIPS  BRANCH=431
//**  DATE=MARCH 1, 1974
//**
//TAB  EXEC PGM=UTTABGEN,REGION=&RGN
//STEPLIB DD  DISP=SHR,DSN=&PTFJOBL
//          DD  DISP=SHR,DSN=&JOBLIB
//SYSOUT DD  SYSOUT=(&A,&CL)
//SYSUDUMP DD SYSOUT=(&A,&CL1)
//SYSPRINT DD SYSOUT=(&A,&CL2)
//SORTLIB DD  DISP=SHR,DSNAME=SYS1.SORTLIB
//SORTWK01 DD  SPACE=(CYL,(&SORTSP),,CONTIG),UNIT=&STG
//SORTWK02 DD  SPACE=(CYL,(&SORTSP),,CONTIG),UNIT=(&STG,SEP=SORTWK01),
//          SEP=SORTWK01
//SORTWK03 DD  SPACE=(CYL,(&SORTSP),,CONTIG),UNIT=(&STG,SEP=SORTWK02),
//          SEP=SORTWK02
//SORTWK04 DD  SPACE=(CYL,(&SORTSP),,CONTIG),UNIT=(&STG,SEP=(SORTWK01,
//          SORTWK03)),SEP=(SORTWK01,SORTWK03)
//SORTWK05 DD  SPACE=(CYL,(&SORTSP),,CONTIG),UNIT=(&STG,SEP=(SORTWK02,
//          SORTWK04)),SEP=(SORTWK02,SORTWK04)
//SORTWK06 DD  SPACE=(CYL,(&SORTSP),,CONTIG),UNIT=(&STG,SEP=(SORTWK01,
//          SORTWK03,SORTWK05)),SEP=(SORTWK01,SORTWK03,SORTWK05)
//SYSLMOD DD  DSNAME=&LIB.L,VOLUME=&VLIB,UNIT=&ULIB,SPACE=(CYL,&LIBSP),
//          DISP=(&LIBDISP,KEEP),DCB=(RECFM=U,BLKSIZE=&BLK)

```

```

00000100
00000200
00000300
00000310
00000400
00000500
00000600
00000700
00000800
00000900
00001000
00001010
00001100
00001200
00001300
00001400
00001500
X00001600
00001700
X00001800
00001900
X00002000
00002100
X00002200
00002300
X00002400
00002500
X00002600
00002700

```

## NIPS 360 FFS

## PROCEDURES

```

//XTP      PROC  A=A,                                00000100
//          CL=' ',CL1=' ',CL2=U,CL3=C,CL4=S,CL5=T,    00000200
//          JOBLIB='FFS.JOBLIB',                      00000300
//          LIB='DUMMY.FILE',                          00000400
//          LIB1='DUMMY.FILE',                        00000500
//          PTFJOBL='PTF.JOBLIB',                     00000600
//          RESTART=REFORMAT,                          00000700
//          RGN=82K,                                    00000800
//          STG=NIPW,TPIMQ=TPIMQ,                      00000900
//          TPDUMP='SYSOUT=(A,,)',                    00001000
//          ULIB=2314,ULIB1=2314,                      00001100
//          UOMQ=2314,                                  00001200
//          VOMQ='REF=*INMSGQ',                        00001300
//          VLIB=,VLIB1=,                              00001400
//          XINDEX='DUMMY.FILE',                       00001500
//          XUNIT=2314,XVOL=                           00001600
// *                                                  00001700
// * CHARLES W. HICKISCH MAJOR,USA  PRJ CODE=763NIPS  BRANCH=431 00001800
// * DATE=MARCH 1, 1974                                00001900
// *                                                  00002000
//TPMONSUP   EXEC PGM=UTTPDRVR,REGION=&RGN,PARM=&RESTART 00002100
//STEPLIB DD DSN=&PTFJOBL,DISP=SHR                    00002200
//          DD DSN=&JOBLIB,DISP=SHR                    00002300
//SYSUT1 DD SPACE=(TRK,(0,5)),UNIT=&STG                00002400
//SYSUT2 DD SPACE=(TRK,(0,20)),UNIT=&STG                00002500
//SYSUT3 DD SPACE=(TRK,(0,5)),UNIT=(&STG,SEP=SYSUT1)    00002600
//SYSUT4 DD SPACE=(CYL,(0,1)),UNIT=&STG                00002700
//SLIB DD DSN=&LIB.L,UNIT=&ULIB,VOLUME=&VLIB,DISP=SHR    00002800
//          DD DSN=&LIB1.L,UNIT=&ULIB1,VOLUME=&VLIB1,DISP=SHR 00002900
//          DD DSN=&JOBLIB,DISP=SHR                    00003000
//DATAFILE DD DISP=SHR,DSN=DUMMY.FILE,UNIT=(,P,DEFER)  00003100
//DATAFILE DD DSN=DUMMY.FILE67,UNIT=(NIPW,2),SPACE=(TRK,(0)),DISP=NEW 00003200
//DATAFIL2 DD DISP=SHR,DSN=DUMMY.FILE,UNIT=(2314,P,DEFER) 00003300
//DATAFIL3 DD DISP=SHR,DSN=DUMMY.FILE,UNIT=(2314,P,DEFER) 00003400
//SAMFILE DD DISP=SHR,DSN=DUMMY.FILES,UNIT=(2314,P,DEFER) 00003500
//EDCONSOLE DD SPACE=(TRK,(5,,4)),UNIT=&STG            00003600
//EDITLIB DD DSN=DUMMY.FILEL,DISP=SHR                 00003700
//INMSGQ DD DISP=SHR,DSN=&TPIMQ                       00003800
//OMSGQ DD DISP=SHR,DSN=&TPIMQ                        00003900
//OUTMSGQ DD SPACE=(TRK,0),VOLUME=&VOMQ,UNIT=&UOMQ      00004000
//AMSGQ DD UNIT=&STG,SPACE=(TRK,0)                     00004100
//SDCONSOLE DD UNIT=&STG,SPACE=(2300,(10,,4))           00004200
//SDKNSET DD SPACE=(CYL,(1,1)),UNIT=&STG,DCB=(RECFM=F,BLKSIZE=1004) 00004300
//STATRECS DD SYSOUT=(&A,&CL)                         00004400
//SYSLMOD DD SPACE=(TRK,(20,,8)),UNIT=&STG,DCB=FFS.JOBLIB, 00004500
// LABEL=EXPDT=66366                                00004600
//SYSPRINT DD SYSOUT=(&A,&CL)                          00004700
//SYSABEND DD SYSOUT=(&A,&CL1)                        00004800
//TPDUMP DD SYSOUT=(&A,&CL1)                          00004900
//EDITDUMP DD SYSOUT=(&A,&CL1)                        00005000
//SYSONLIN DD &TPDUMP                                00005100
//XINDEX DD DSN=&XINDEX.X,UNIT=&XUNIT,VOL=&XVOL,DISP=SHR 00005200
//SHARDASD DD DSN=NIPS.TPJOBQ,DISP=(MOD,KEEP)          00005300
//SYSIN DD DUMMY                                       00005400
//U DD SYSOUT=(&A,&CL2)                                00005500
//C DD SYSOUT=(&A,&CL3)                                00005600
//S DD SYSOUT=(&A,&CL4)                                00005700
//T DD SYSOUT=(&A,&CL5)                                00005800
//SUBFILE DD UNIT=&STG,SPACE=(CYL,(0,1,10))            00005900

```

CH-1



## NIPS 360 FFS

## PROCEDURES

//XTPMON	PROC	A=A,CL=' ',CL1=' ',JOB LIB='FFS.JOB LIB',	00000100
//		N2260=2,OMIT1=NULLUNT1,RESTART=REFORMAT,	00000200
//		PTFJOB L='PTF.JOB LIB',	00000210
//		TPIMQ=TPIMQ,UDD5=2E0,UDD6='2260-1',	00000300
//		UOMQ=,VOMQ='REF=*.INMSGQ'	00000400
/**			00000500
/**	CHARLES W. HICKISCH	MAJOR,USA PRCJ CODE=763NIPS BRANCH=431	00000600
/**	DATE=MARCH 1, 1974		00000700
/**			00000800
//TPMON	EXEC	PGM=TPMON,PARM=&RESTART,TIME=1439	00000900
//STEPLIB	DD	DISP=SHR,DSN=&PTFJOB L	00001000
//	DD	DISP=SHR,DSN=&JOB LIB	00001010
//STATRECS	DD	SYSOUT=( &A,&CL)	00001100
//SYSUDUMP	DD	SYSOUT=( &A,&CL1)	00001200
//INMSGQ	DD	DISP=SHR,DSNAME=&TPIMQ	00001300
//OUTMSGQ	DD	SPACE=( TRK,0),VOLUME=&VOMQ,UNIT=&UOMQ	00001400
//NOUNIT1	DD	DDNAME=&OMIT1	00001500
//DD5	DD	UNIT=&UDD5	00001600
//DD6	DD	UNIT=( &UDD6,&N2260)	00001700

## NIPS 360 FFS

## PROCEDURES

```

//XTPSUP PRC A=A,                                00000100
//      CL=' ',CL1=' ',CL2=U,CL3=C,CL4=S,CL5=T,    00000200
//      JOBLIB='FFS.JOBLIB',                      00000300
//      LIB='DUMMY.FILE',                          00000400
//      LIB1='DUMMY.FILE',                         00000500
//      PTFJOB='PTF.JOBLIB',                      00000600
//      RGN=82K,                                    00000700
//      STG=NIPW,                                    00000800
//      TPDUMP='SYSOUT=(A,)',                     00000900
//      ULIB=2314,ULIB1=2314,                      00001000
//      VLIB=,VLIB1=,                              00001100
//      XINDEX='DUMMY.FILE',                      00001200
//      XUNIT=2314,XVOL=                          00001300
//**                                                00001400
//** CHARLES W. HICKISCH MAJOR,USA PRCJ CODE=763NIPS BRANCH=431 00001500
//** DATE=MARCH 1, 1974                          00001600
//**                                                00001700
//TPSUP EXEC PGM=TPSUP,REGION=&RGN,TIME=1435      00001800
//STEPLIB DD DSN=&PTFJOB,DISP=SHR                 00001900
//      DD DSN=&JOBLIB,DISP=SHR                   00002000
//SYSUT1 DD SPACE=(TRK,(0,5)),UNIT=&STG           00002100
//SYSUT2 DD SPACE=(TRK,(0,20)),UNIT=&STG          00002200
//SYSUT3 DD SPACE=(TRK,(0,5)),UNIT=(&STG,SEP=SYSUT1) 00002300
//SYSUT4 DD SPACE=(CYL,(0,1)),UNIT=&STG           00002400
//SLIB DD DSN=&LIB.L,UNIT=&ULIB,VOLUME=&VLIB,DISP=SHR 00002500
//      DD DSN=&LIB1.L,UNIT=&ULIB1,VOLUME=&VLIB1,DISP=SHR 00002600
//      DD DSN=&JOBLIB,DISP=SHR                   00002700
//DATAFILE DD DISP=SHR,DSNAME=DUMMY.FILE,UNIT=(,P,DEFER) 00002800
//DATAFIL1 DD DSN=DUMMY.FILE67,UNIT=(NIPW,2),SPACE=(TRK,(0)),DISP=NEW 00002900
//DATAFIL2 DD DISP=SHR,DSN=DUMMY.FILE,UNIT=(2314,P,DEFER) 00003000
//DATAFIL3 DD DISP=SHR,DSN=DUMMY.FILE,UNIT=(2314,P,DEFER) 00003100
//SAMFILE DD DISP=SHR,DSN=DUMMY.FILES,UNIT=(2314,P,DEFER) 00003200
//EDCONSOL DD SPACE=(TRK,(5,,4)),UNIT=&STG        00003300
//EDITLIB DD DSN=DUMMY.FILEL,DISP=SHR            00003400
//INMSGQ DD DISP=SHR,DSNAME=TPIMQ                 00003500
//OMSGQ DD DISP=SHR,DSNAME=TPIMQ                  00003600
//AMSGQ DD UNIT=&STG,SPACE=(TRK,0)                 00003700
//SDCONSOL DD UNIT=&STG,SPACE=(2300,(10,,4))       00003800
//SDKNSET DD SPACE=(CYL,(1,1)),UNIT=&STG,DCB=(RECFM=F,BLKSIZE=1004) 00003900
//SYSLMOD DD SPACE=(TRK,(20,,8)),UNIT=&STG,CCB=FFS.JOBLIB, 00004000
// LABEL=EXPDT=66366                              00004100
//SYSPRINT DD SYSOUT=(&A,&CL)                      00004200
//SYSABEND DD SYSOUT=(&A,&CL1)                     00004300
//TPDUMP DD SYSOUT=(&A,&CL1)                       00004400
//EDITDUMP DD SYSOUT=(&A,&CL1)                     00004500
//SYSONLIN DD &TPDUMP                             00004600
//XINDEX DD DSN=&XINDEX.X,UNIT=&XUNIT,VOL=&XVOL,DISP=SHR 00004700
//SHARDASD DD DSN=NIPS.TPJOBQ,DISP=(MOD,KEEP)     00004800
//SYSIN DD DUMMY                                  00004900
//U DD SYSOUT=(&A,&CL2)                            00005000
//C DD SYSOUT=(&A,&CL3)                            00005100
//S DD SYSOUT=(&A,&CL4)                            00005200
//T DD SYSOUT=(&A,&CL5)                            00005300
//SUBFIL DD UNIT=&STG,SPACE=(CYL,(0,1,10))         00005400

```

## NIPS 360 FFS

## PROCEDURES

```

//XTRDISK      PROC  A=A,CL=',',CL1=',',JOB LIB='FFS.JCBLIB',      00000100
//              LAB=BLP,RGN=60K,SAM=,USAM='(TAPE9,,DEFER)',      00000110
//              PTFJOBL='PTF.JOBLIB',      00000200
//              VSAM=,XFDISP=SHR,XFNAME=,XFUNIT=2314,XFVOL=      00000300
//*****      00000400
//*      * 00000500
//*      THIS PROC IS USED TO TRANSFER A DISK-RESIDENT INDEX DATA SET * 00000600
//* TO TAPE. THIS OPERATION CONDENSES THE INDEX DATA SET. THE TAPE * 00000700
//* SO CREATED IS A SEQUENTIAL DATA SET CONSISTING OF VARIABLE LENGTH * 00000800
//* BLOCKED RECORDS THAT CONTAIN THE SOURCE DATA AND CONTROL * 00000900
//* INFORMATION FOR SUBSEQUENTLY RECONSTRUCTING THE SOURCE DATA. * 00001000
//*      * 00001100
//*      //STEPNAME EXEC XTRDISK,XFNAME=WW,SAM=XX,XFVOL=YY,VSAM=ZZ * 00001200
//*      * 00001300
//*      CHARLES W. HICKISCH MAJOR,USA PRGJ CODE=763NIPS BRANCH=431 00001400
//*      DATE=MARCH 1, 1974 00001500
//*      * 00001600
//*****      00001700
//*      * 00001800
//XTR          EXEC PGM=UTNDXTRF,PARM=CISK,REGION=&RGN      00001900
//STEPLIB DD    DISP=SHR,DSN=&PTFJOBL      00002000
//          DD    DISP=SHR,DSN=&JOBLIB      00002010
//INDEXDAM DD    DSN=&XFNAME,UNIT=&XFUNIT,VOL=&XFVOL,DISP=&XFDISP 00002100
//INDEXPRT DD    SYSOUT=(&A,&CL1)      00002200
//INDEXSAM DD    DSN=&SAM,UNIT=&USAM,VOL=&VSAM,DISP=(NEW,KEEP), *00002300
//          LABEL=(,&LAB)      00002400
//SYSPRINT DD    SYSOUT=(&A,&CL)      00002500
//SYSUDUMP DD    SYSOUT=(&A,&CL1)      00002600

```

## NIPS 360 FFS

## PROCEDURES

```

//XRTAPE      PROC  A=A,CL=',' ,CL1=',' ,JOB LIB='FFS.JOB LIB',      00000100
//            LAB=BLP,NBRBLK=50,RGN=60K,SAM=,      00000200
//            PTFJOB LIB='PTF.JOB LIB',      00000210
//            USAM='(TAPE9,,DEFER)',VSAM=,XFDISP=CLD,      C00000300
//            XTDISP='(NEW,KEEP)',XTNAME=,XTUNIT=2314,XTVOL=      00000400
//*****      00000500
//*          * 00000600
//*          * 00000700
//*          * 00000800
//*          * 00000900
//*          * 00001000
//*          * 00001100
//*          * 00001200
//*          * 00001300
//*          * 00001400
//*          * 00001500
//*****      00001600
//XTR          EXEC PGM=UTNDXTR,PARM=TAPE,REGION=&RGN      00001700
//STEPLIB DD   DISP=SHR,DSN=&PTFJOB LIB      00001900
//          DD   DISP=SHR,DSN=&JOB LIB      00001810
//INDEXPRT DD   SYSOUT=(&A,&CL)      00001900
//INDEXSAM DD   DSN=&SAM,UNIT=&USAM,VOL=&VSAM,DISP=&XFDISP,      C00002000
//          LABEL=(,&LAB)      00002100
//SYSPRINT DD   SYSOUT=(&A,&CL)      00002200
//SYSUDUMP DD   SYSOUT=(&A,&CL1)      00002300
//XINDEX DD   DSN=&XTNAME,UNIT=&XTUNIT,VOL=&XTVOL,      *00002400
//          DISP=&XTDISP,SPACE=(560,&NBRBLK),      C00002500
//          DCB=1BLKSIZE=560,RECFM=F,KEYLEN=4,DSORG=DA)      00002600

```



## NIPS 360 FFS

## PROCEDURES

```

//XUTFSCAN      PROC  A=A,BSZFILE=,                                00000100
//              CL=',' ,CL1=',' ,DEN=,ISAM='DUMMY.FILE',          00000200
//              JOBLIB='FFS.JOBLIB',LAB=SL,LIB='DUMMY.FILE',      00000300
//              LIB1='DUMMY.FILE',RGN=98K,SAM='DUMMY.FILE',        00000400
//              PTFJOBL='PTF.JOBLIB',                             00000410
//              SORTSP=10,STG=NIPW,TBLK=500,TDISP=MOD,            00000500
//              TRANS='DUMMY.FILE',TRANSP=200,TRCH=,UISAM='(2314,P)', 00000600
//              ULIB=2314,ULIB1=2314,USAM='(TAPE9,,DEFER)',        00000700
//              UTRANS=2314,VISAM=,VLIB=,VLIB1=,VSAM=,VTRANS=      00000800
//**                                                     00000900
//**  CHARLES W. HICKISCH  MAJOR,USA  PRJ CODE=763NIPS  BRANCH=431 00001000
//**  DATE=MARCH 1, 1974                                     00001100
//**                                                     00001200
//UTFLDSCN EXEC PGM=UTFLDSCN,REGION=&RGN                        00001300
//STEPLIB DD DISP=SHR,DSN=&PTFJOBL                             00001400
//          DD DISP=SHR,DSN=&JOBLIB                             00001410
//SYSPRINT DD SYSCUT=(&A,&CL)                                   00001500
//SYSUDUMP DD SYSOUT=(&A,&CL1)                                   00001600
//SYSCUT DD SYSOUT=(&A,&CL1)                                    00001700
//DATAFILE DD DSNNAME=&ISAM,UNIT=&UISAM,VOLUME=&VISAM,DISP=SHR   00001800
//SORTLIB DD DSNNAME=SYS1.SORTLIB,DISP=SHR                     00001900
//SORTIN DD UNIT=&STG,SPACE=(TRK,(&TRANSP,150))                 00002000
//SORTWK01 DD SPACE=(CYL,(&SORTSP),,CONTIG),UNIT=(&STG,SEP=SORTIN) 00002100
//SORTWK02 DD SPACE=(CYL,(&SORTSP),,CONTIG),                     00002200
//          UNIT=(&STG,SEP=SORTWK01),SEP=SORTWK01               00002300
//SORTWK03 DD SPACE=(CYL,(&SORTSP),,CONTIG),                     00002400
//          UNIT=(&STG,SEP=SORTWK02),SEP=SORTWK02               00002500
//SORTWK04 DD SPACE=(CYL,(&SORTSP),,CONTIG),                     00002600
//          UNIT=(&STG,SEP=(SORTWK01,SORTWK03)),                 00002700
//          SEP=(SORTWK01,SORTWK03)                               00002800
//SORTWK05 DD SPACE=(CYL,(&SORTSP),,CONTIG),                     00002900
//          UNIT=(&STG,SEP=(SORTWK02,SORTWK04)),                 00003000
//          SEP=(SORTWK02,SORTWK04)                               00003100
//SORTWK06 DD SPACE=(CYL,(&SORTSP),,CONTIG),                     00003200
//          UNIT=(&STG,SEP=(SORTWK01,SORTWK03,SORTWK05)),        00003300
//          SEP=(SORTWK01,SORTWK03,SORTWK05)                     00003400
//ISAMWORK DD UNIT=&STG,DCB=DSORG=IS,SPACE=(CYL,(10))            00003500
//SAMFILE DD DSNNAME=&SAM.S,UNIT=&USAM,VOLUME=&VSAM,DISP=(SHR,KEEP), 00003600
//          LABEL=(, &LAB),DCB=(RECFM=VB,LRECL=1000,BLKSIZE=&BSZFILE, 00003700
//          TRCH=&TRCH,DEN=&DEN)                                   00003800
//SYSUT2 DD DSNNAME=&LIB.L,UNIT=&ULIB,VOLUME=&VLIB,DISP=SHR        00003900
//          DD DSNNAME=&LIB1.L,UNIT=&ULIB1,VOLUME=&VLIB1,DISP=SHR 00004000
//TRANST DD DSNNAME=&TRANS,UNIT=&UTRANS,VOLUME=&VTRANS,DISP=&TDISP, 00004100
//          DCB=(RECFM=FB,LRECL=50,BLKSIZE=&TBLK)               00004200

```

NIPS 360 FFS PROCEDURES

//XUTSOURC	PROC CL='',CL1='',JOBLIB='FFS.JOBLIB',	00000100
//	PTFJOBL='PTF.JOBLIB',	00000110
//	NAME=,SDISP=SHR,SOURCL=,USOURCL=2314,VSCOURCL=	00000200
/**		00000300
/**	CHARLES W. HICKISCH MAJOR,USA PRJ CODE=763NIPS BRANCH=431	00000400
/**	DATE=MARCH 1, 1974	00000500
/**		00000600
//SOURC EXEC PGM=UTSOURC		00000700
//STEPLIB DD DISP=SHR,DSN=&PTFJOBL		00000800
// DD DISP=SHR,DSN=&JOBLIB		00000810
//SOURCPRT DD SYSOUT=(A,&CL)		00000900
//SOURCLIB DD DISP=&SDISP,DSN=&SOURCL.L,VOL=&VSOURCL,UNIT=&USOURCL		00001000
//SYSUDUMP DD SYSOUT=(A,&CL1)		00001100
//SYSIN DD DISP=SHR,DSN=&SOURCL.L(&NAME),UNIT=&USOURCL,VOL=&VSOURCL		00001200

## NIPS 360 FFS

## PROCEDURES

//X1410CON	PROC A=A,CL=' ',CL1=' ',ISAM='DUMMY.FILE',	00000100
//	JOBLIB='FFS.JOBLIB',RGN=60K,SAM=DUMMY,	00000200
//	PTFJOBL='PTF.JOBLIB',	00000210
//	UISAM='(2314,P)',U1410='(TAPE7,,DEFEP)',	00000300
//	VISAM=,V1410=	00000400
/**		00000500
/**	CHARLES W. HICKISCH MAJOR,USA PROJ CODE=763NIPS BRANCH=431	00000600
/**	DATE=MARCH 1, 1974	00000700
/**		00000800
//GO EXEC	PGM=UTDATAC,REGION=&RGN	00000900
//STEPLIB DD	DISP=SHR,DSN=&PTFJOBL	00001000
// DD	DISP=SHR,DSN=&JOBLIB	00001010
//SYSPRINT DD	SYSOUT=(&A,&CL)	00001100
//SYSOUT DD	SYSOUT=(&A,&CL)	00001200
//SYSUDUMP DD	SYSOUT=(&A,&CL1)	00001300
//DATAFILE DD	DISP=SHR,DSN=&ISAM,UNIT=&UISAM,VOLUME=&VISAM	00001400
//NEWFILE DD	DSNAME=&SAM.XXX,UNIT=&U1410,VOLUME=&V1410,LABEL=(,NSL),	X00001500
//	DCB=(BLKSIZE=2704,TRTCH=ET,DEN=1),DISP=(,KEEP)	00001600

## NIPS 360 FFS

## PROCEDURES

```

//X360CON      PROC  A=A,CL=' ',CL1=' ',ISAM='DUMMY.FILE',      00000100
//              JOBLIB='FFS.JOBLIB',LAB=BLP,NDISP=PASS,          00000200
//              ODISP=DELETE,RGN=60K,SAM=,UISAM='(2314,P)',      00000300
//              PTFJOBL='PTF.JOBLIB',                            00000310
//              USAM='(TAPE9,,DEFER)',U1410='(TAPE7,,DEFER)',    00000400
//              VISAM=,VSAM=,V1410=                                00000500
//*                                                     00000600
//*  CHARLES W. HICKISCH  MAJOR,USA  PROJ CODE=763NIPS  BRANCH=431  00000700
//*  DATE=MARCH 1, 1974                                           00000800
//*                                                     00000900
//GO EXEC PGM=UTDATAC,REGION=&RGN                                00001000
//STEPLIB DD DISP=SHR,DSN=&PTFJOBL                                00001100
//          DD DISP=SHR,DSN=&JOBLIB                                00001110
//SYSUDUMP DD SYSOUT=(&A,&CL1)                                    00001200
//SYSPRINT DD SYSOUT=(&A,&CL)                                     00001300
//SYSOUT DD SYSOUT=(&A,&CL)                                       00001400
//DATAFILE DD DCR=DSORG=IS,DISP=(SHR,&ODISP,KEEP),DSNAME=&ISAM,  X00001500
//              UNIT=&UISAM,VOLUME=&VISAM                          00001600
//FILE1410 DD DISP=(OLD,PASS),LABEL=(2,BLP),VOLUME=&V1410,DSNAME=&SAM, *00001700
//              UNIT=&U1410,DCB=(DEN=1,TRTCH=ET,BLKSIZE=2704)    00001800
//NEWFILE DD DISP=(, &NDISP),UNIT=&USAM,LABEL=(, &LAB),          X00001900
//              DSNAME=&SAM.S,VOLUME=&VSAM                          00002000

```





DEFENSE COMMUNICATIONS AGENCY

COMMAND AND CONTROL  
TECHNICAL CENTER  
WASHINGTON, D. C. 20301

10 June 1976

IN REPLY  
REFER TO:

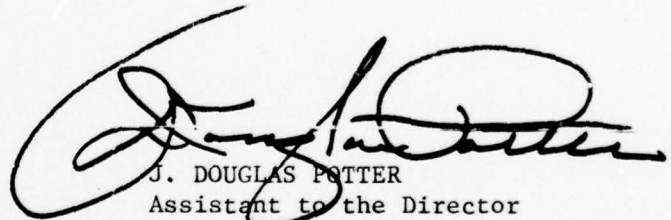
TO: DISTRIBUTION

SUBJECT: Change 2 to CSM UM 15-74, Job Preparation,  
Volume VIII, dated 15 October 1974

1. Insert the enclosed change pages and destroy the replaced pages according to applicable security regulations.
2. A list of Effective Pages to verify the accuracy of this manual is enclosed. This list should be inserted before the title page.
3. When this change has been posted, make an entry in the Record of Changes on the inside cover.

FOR THE DIRECTOR:

19 Enclosures  
Change 2 pages

  
J. DOUGLAS POTTER  
Assistant to the Director  
for Administration



EFFECTIVE PAGES - 10 June 1976

This list is used to verify the accuracy of CSM UM 15-74, Volume VIII, after change 2 pages have been inserted. Original pages are indicated by the letter O, change 2 by the numeral 2.

<u>Page No.</u>	<u>Change No.</u>
Title Page	2
ii - iii	O
iv - v	2
vi - vii	O
1 - 42	O
43 - 44	2
45 - 52	O
53 - 54	2
54.1 - 54.2	2
55 - 64	O
65 - 66	2
66.1 - 66.2	2
67 - 76	O
77 - 78	2
79 - 100	O
101 - 102	2
103 - 150	O
151 - 152	2
153 - 161	O

CCMMAND AND CONTROL TECHNICAL CENTER  
Computer System Manual Number CSM UM 15-74

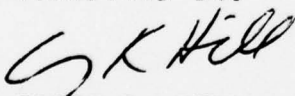
10 June 1976

NMCS INFORMATION PROCESSING SYSTEM  
360 FORMATTED FILE SYSTEM (NIPS 360 FFS)

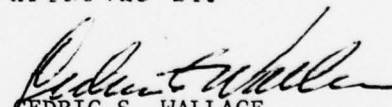
Users Manual

Volume VIII - Job Preparation

SUBMITTED BY:

  
CRAIG K. HILL  
Captain USA  
CCTC Project Officer

APPROVED BY:

  
CEDRIC S. WALLACE  
Captain, USN  
Deputy Director, NMCS ADP

This document has been approved for public release and sale;  
its distribution is unlimited.

CH-2

Section	Page
3.8	RASP Procedure (XPASP)..... 39
3.8.1	Sample Job Setup..... 40
#3.9	SAM to ISAM Procedure (XSTOIS)..... 43
3.9.1	Sample Job Setup..... 44
3.10	Subroutine Loader Procedure (XSUBLDR)..... 44
3.10.1	Sample Job Setup..... 44
3.11	Table Generation Procedure (XTABGEN)..... 45
3.11.1	Sample Job Setup..... 45
3.12	Source Language Library Procedure (XUTSOURC) 45
3.12.1	Sample Job Setup..... 45
3.13	1410 to 360 Conversion (X360CON)..... 46
3.13.1	Sample Job Setup..... 46
3.14	360 to 1410 Conversion (X1410CON)..... 47
3.14.1	Sample Job Setup..... 47
3.15	List Logic Statement/Report Names Procedure (XDMLIB)..... 47
3.15.1	Sample Job Setup..... 48
3.16	Subroutine Check Procedure (XSUBCHK)..... 48
3.16.1	Sample Job Setup..... 48
3.17	Classification Change Procedure (XCLASS)..... 48
3.17.1	Sample Job Setup..... 49
3.18	Data Field Scan Procedure (XUTFSCAN)..... 50
3.18.1	Sample Job Setup..... 50
3.19	Index Specification Procedure (XSP)..... 51
3.19.1	Sample Job Setup..... 51
3.20	Unload Index Data Set Procedure (XTRDISK)..... 52
3.20.1	Sample Job Setup..... 52
3.21	Load Index Data Set Procedure (XTRTAPE)..... 53
3.21.1	Sample Job Setup..... 53
3.22	Keyword Analysis (XKA)..... 53
3.22.1	Sample Job Setup..... 53
3.23	Dictionary Maintenance Utility (XKM)..... 54
3.23.1	Sample Job Setup..... 54
#3.24	Format Definition Translator Procedure (XUTCDE)..... 54
#3.24.1	Sample Job Setup..... 54.1
4	SUPPLEMENTARY PROCEDURES DESCRIPTIONS..... 55
4.1	Restore Answers Procedure (XRESTANS)..... 55
4.1.1	Sample Job Setup..... 55
4.2	Restore Library Procedure (XRESTLIB)..... 56
4.2.1	Sample Job Setup..... 56
4.3	Save Answers Procedure (XSAVEANS)..... 57
4.3.1	Sample Job Setup..... 57
4.4	Save Library Procedure (XSAVELIB)..... 57
4.4.1	Sample Job Setup..... 58
5	PRODUCTION PROCEDURES DESCRIPTION..... 59
5.1	PM Production Procedure (XFMEX)..... 59
5.2	OP Production Procedure (XOPEX, XOPSDX)..... 59



Section	Page
5.3	Compression and Compaction of Data Records... 60
5.4	RASP Production Procedure (XRASPEX) ..... 61
APPENDIX	
# A	SYMBOLIC PARAMETER DEFINITIONS..... 62
B	INPUT SOURCE DDNAME..... 71
C	PROCEDURES DESCRIPTIONS..... 71
D	NIPS PROCEDURE RETURN CODES..... 75
# E	NIPS PROCEDURE DD STATEMENT USAGE..... 77
# F	PROCEDURE LISTINGS..... 108

XCLASS  
 XDMPLIE  
 XFM  
 XFMEX  
 XFR  
 XFS  
 XISTOS  
 XKA  
 XKM  
 XCP  
 XOPEX  
 XCPSD  
 XOPSDEX  
 XQRTQDF  
 XQUIP  
 XQUIPSD  
 XRASPEX  
 XRESTANS  
 XRESTLIB  
 XSAVEANS  
 XSAVELIB  
 XSP  
 XSTOIS  
 XSUBCHK  
 XSUBLDR  
 XTABGEN  
 XTP  
 XTPMON  
 XTPSUP

## JOB PREPARATION

be performed with library TEST360L. TRAINERS is an indexed file.

```
//RET1 EXEC XRASP,ISAM=TEST360,SAM=TRAINER,LIB=TEST360, X
//          LIB1=TRAINER,LIBDISP=OLD,XINDEX=TRAINER
//RASP.SYSIN DD *
//          (RASP CONTROL AND SOURCE LANGUAGE STATEMENTS)
/*
```

### 3.9 SAM TO ISAM Procedure (XSTOIS)

This procedure is used to load a tape data file (SAM) to disk (ISAM). Space is allocated separately for INDEX, PRIME, and OVFLOW areas. The procedure defaults these areas to 1, 40, and 5 cylinders, respectively. The disposition of the new disk file defaults to KEEP.

If the input is SAM file on unlabeled magnetic tape and its block size is not 1,004 bytes, its block size must be specified with the BSZFILE symbolic parameter.

The BSZNEWF symbolic parameter is used to specify a new block size for the ISAM file. If it is not used, the ISAM block size will be the same as the input SAM block size.

# The procedure also can be used to create an ISAM file in the compression/compaction form or to re-create a standard form ISAM file. The PARM parameter can be used for specifying the option. The default value is the form of the input file. The options are:

PARM=COMPRESS	for compression only
PARM=CCMPACT	for compaction only
PARM='CCMPRESS,CCMPACT'	for both compression and compaction
PARM=EXPAND	for reversing the compression/compaction process to produce standard form data records

## JOB PREPARATION

# All unused PRIME space will be filled with system generated PAD records. The PARM option NOPAD may be used if this feature is not desired, e.g.:

PARM=NOPAD

### 3.9.1 Sample Job Setup

The following JCL could be used to load the TEST360S file from tape-to-disk. It is desired to increase the PRIME area to 50 cylinders and catalog the new data set. The SAM file is not cataloged.

```
//STOIS EXEC XSTCIS,SAM=TEST360,PRIME=50, X
// VSAM='SER=MYTAPE',ISAM=TEST360, X
// VISAM='SER=MYPACK',NDISP=CATLG
/*
```

### 3.10 Subroutine Loader Procedure (XSUBLDR)

This procedure is used to place user subroutines on the file library. The two inputs are the user subroutine in load module form on a temporary or permanent data set and a control card. The MODLIB parameter is used to specify the name of the data set containing the input subroutine. It defaults to &TEMP which is the name of the data set containing the subroutine if the ASMFCL procedure has been used in a prior step of the job to assemble and link edit the subroutine.

#### 3.10.1 Sample Job Setup

The following JCL could be used to assemble, link-edit, and add a subroutine to the TEST360L library which is cataloged. The subroutine is named DTGIS.

Note: The MODLIB DSNAME is not specified on the EXEC card since it defaults to the DSNAME of the assembler job step output (&TEMP).

```
//ASML EXEC ASMFCL
//ASM.SYSIN DD *
```

SUBROUTINE SOURCE DECK (DTGIS)

/\*

## JOB PREPARATION

```
//STEPNAME EXEC XTRDISK,STAT=YES
//                      SAM=INDEXSAM,VSAM='SER=MYTAPE',
//                      XFNAME=TEST360X,XFVOL='SER=MYPACK',
/*
```

### 3.21 Load Index Data Set Procedure (XTRTAPE)

This procedure is used to reconstruct a disk-resident direct access organization Index Data Set from a previously unloaded sequential access organization data set. The XFNAME and XFVOL symbolic parameters must be supplied for the input tape. The XTNAME and XTVOL symbolic parameters must be supplied for the Output Index Data Set on disk.

#### 3.21.1 Sample Job Setup

The following JCL would load the TEST360X Index Data Set to a 2314 disk pack from a nine-track unlabeled tape.

```
//STEPNAME EXEC XTRTAPE,
//                      XTNAME=TEST360X,XTVOL='SER=MYPACK',
//                      SAM=INDEXSAM,VSAM='SER=MYTAPE'
/*
```

### 3.22 Keyword Analysis (XKA)

This procedure is used to obtain printed listing of text, non-keywords, and keywords that occur in a data base without updating the index data set. No DD card overrides are necessary, but a DD statement named SYSIN must always be included to identify the user's input. A file name must be specified using either the ISAM or SAM symbolic parameter. If a user scan subroutine, stop word table, or dictionary is required, the library name must be specified with the LIB symbolic parameter.

#### 3.22.1 Sample Job Setup

The following JCL would be used to analyze keyword fields in the TEST360 ISAM data base:



## JOB PREPARATION

```
//STEPNAME EXEC XKA,ISAM=TEST360,LIB=TEST360,  
//          VISAM='SER=MYPACK',VLIB='SER=MYPACK'  
//XKA.SYSIN DD *  
          (user supplied control statements)  
/*
```

### 3.23 Dictionary Maintenance Utility (XKM)

This procedure is used to generate or update keywords, stop word tables and dictionaries. More than one stop word table or dictionary may be created or updated in one run provided that all are members in one library (specified by the LIB parameter), stop word table and dictionary name specifications, maintenance commands, and data are accepted through the SYSIN device.

#### 3.23.1 Sample Job Setup

The following JCL would be used to maintain any or all stop word tables and dictionaries in the TEST360 library.

```
// EXEC XKM,LIB=TEST360  
//SKM.SYSIN DD *  
      Table or Dictionary Names  
      Maintenance Commands  
      Data (keyword specs)  
/*
```

### #3.24 Format Definition Translator Procedure (XUTODE)

This procedure is used to place format definitions on the user library. The input definition statements may be in punched cards or in card image records stored in a library. More than one format definition may be added to the library at a time.

#### #3.24.1 Sample Job Setup

The following JCL would be used to add a format definition in punched cards to the user library, TEST360.

# JOB PREPARATION

```
// EXEC XUTODE,LIB=TEST360,VLIE='SER=MYPACK'  
//SYSIN DD *  
    Definition source statement cards.
```

```
/*
```

To add a format to the TEST360 library when the definition source statements reside in a library (MYLIB), the following JCL would be used.

```
// EXEC XUTODE,LIB=TEST360,VLIE='SER=MYPACK'  
//SYSIN DD DSN=MYLIB (FORMAT1),VOL=SER=MYPACK2,  
//  DISP= (SHR,KEEP),UNIT=2314,  
//  DCB=(RECFM=FB,LRECL=80,BLKSIZE=800  
/*
```

Note that the definition source statements may be on any library, including TEST360. The only restriction is that the source member name is not the same as the name of the format definition.

<u>PARAMETER</u>	<u>DEFINITION</u>
OLDSAM	This defines the name of the old SAM file in the XISTOS procedure.
OLDVSAM	This defines the volume for the old SAM file in the XISTOS procedure.
OSDISP	This defines the disposition of the old SAM file in the XISTOS procedure.
OVFLOW	This defines the number of cylinders allocated to the overflow area for a new ISAM data file.
PRIME	This defines the number of cylinders allocated to the prime area for a new ISAM data file.
#PTFJOBL	This defines the data set name of a partitioned data set containing the PTF load modules.
#PTFJOBM	This defines the data set name of the partitioned data set containing PTF generative code macros.
QDF	This defines the name of the RASP qualified data file if other than a temporary name is required.
QDFSP	This defines the space allocation for the RASP qualified data file.
QDISP	This defines the disposition of the RASP qualified data file and qualified record table.
QRT	This defines the name of the RASP qualified record table.
QRTSP	This defines the space allocation for the RASP qualified record table.
SAM SAM1 SAM2	These define SAM data file names.
SAMIN	This defines the SAM data file name for the XTRDISK procedure when requesting the statistics option.

<u>PARAMETER</u>	<u>DEFINITION</u>
SEQNO	This parameter defines the data set's position with respect to other data sets on the volume.
SDISP	This defines the disposition of the user's source library.
SLAB	This defines the label type for sort work tapes.
SAMOUT	This is used to control the allocation of the FMSAMOUT and FMNDATA data sets in FM.
SORTSP	This defines the cylinder allocation for disk sort work areas.
STAT	This requests the statistics option for for XTRDISK index utility.
STG	This defines the unit type for sort work areas.
SOURCL	This defines the user's source library names.
TBLK	This defines the blocksize for the File Analysis Statistics transaction data set.
TDISP	This defines the disposition for the File Analysis Statistics transaction data set.
TRANS	This defines the File Analysis Statistics transaction data file name.
TRANSP	This defines the space allocation for temporary storage of update transactions with the sort key and logic statement name appended.
TRANTYP	This defines to the FM component the data set containing records to be used as transactions during execution of FR. Default value is ISAM. The value for this parameter is either ISAM or SAM and denotes the access method associated with the file being revised.
TRBUFNO	This defines the number of buffers used by



PARAMETER

DEFINITION

some of the transaction temporary data sets.

TRCH

This specifies the TRTCH DCB parameter for seven-track tapes.

## JOB PREPARATION

SOURCELIB      Defines a library used to store source material.

#SYSUT1        Temporary data set for INDEX Specification statements to be processed by the INDEX Specification processor, IXSP, which builds descriptor (D) records for the file.

SYSIN           Defines the input source statements for PS. This statement must be supplied by the user as //PS.SYSIN DD \*.

### PROCEDURE: XFR (File Revision)

The following DD statements appear in the FR and PRTST steps of this procedure. For a description of all other DD statements, refer to the XPM procedure.

STEPLIB        Defines the NIPS system library.

SYSOUT         Defines a printer output data set for sort messages.

SYSPRINT       Defines a printer output data set.

SYSUDUMP       ABEND dump printer output.

SYSPUNCH       Defines a punch output data set.

OLDFILE        Defines the ISAM data file to be revised.

SAMFILE        Defines the SAM data file to be revised.

DATAFILE       Defines a data set containing the new FFT.

TDDRCDS        Defines a temporary data set for generated TDD cards.

POOLRCDS       Defines a temporary data set for generated POOL statements.

PRVSNA         Defines a temporary data set containing first 24 generated logic statements.

## JOB PREPARATION

FRVSNB	Defines a temporary data set containing second group of 24 generated logic statements if needed.
FRVSNB	Defines the temporary data set containing a third group of 24 generated logic statements if needed.
FRGENCD	Defines the temporary data set. FMS control card for file generation steps; passed to GENT and GEND steps.
SORTLIB	Defines the S/360 Operating System's Sort library.
SORTIN	Defines the temporary data set containing intermediate work records to be sorted.
SORTWK01 through SORTWK06	Defines the work data sets used by SORT. They must all be the same unit type.
SORTOUT	Defines the temporary data set for sorted intermediate work records.
SYSIN	Defines the input source statements for FR. This statement must be supplied by the user as //FR.SYSIN DD *.
TRANSTYP	Defines the DD card to be used for input transaction to FM.
ISAM	Defines to FM where update transactions may be found. Entries in this DD correspond to the OLDFILE DD.
SAM	Defines to FM where update transactions may be found. Entries in this DD correspond to the SAMFILE DD.

## JOB PREPARATION

### Procedure: XUTFSCAN (Field Scan)

SYSPRINT	Defines a printer output data set.
SYSUDUMP	ABEND dump printer output.
DATAFILE	Defines the ISAM data file.
SAMFILE	Defines the SAM data file.
ISAMWORK	Defines temporary ISAM data set if a SAM data file is input.
SYSUT2	Defines partitioned data set containing members to be scanned.
TRANST	Defines the data set for output transactions.
SYSIN	Defines input stream. This card must be supplied by the user as //UTF.SYSIN DD *.
SOURCPRT	Defines a printer output data set.
SOURCLIB	Defines a library used to store source material.
SYSIN	Defines the input to the UTSOURC utility. This is not overridden if a LIST operation is desired. This statement is overridden if library update is to be performed.



## JOB PREPARATION

### Procedure: XSP (Index Specification)

STEPLIB	Defines the NIPS system library.
NEWFILE	Defines the indexed sequential data file created during an FM run.
XINDEX	Defines the Index Data Set.
SORTWK01 through SORTWK06	Defines the S/360 Operating System's Sort work data sets.
SORTLIB	Defines the S/360 Operating System's Sort Library.
SYSOUT	Defines a printer output data set for Sort messages.
SYSPRINT	Defines a printer output data set.
SOURCPRT	Defines a printer output data set for Index Specification messages.
SYSUDUMP	ABEND dump printer output.
SLIB	Defines the user library containing subroutines/tables.
#SAMFILE	Defines a sequential data file
#UTSAMOUT	Defines the updated sequential data file.
ISAMWORK	Defines the temporary ISAM data set containing the FFT during sequential file processing in Index Specification.

## NIPS 360 FFS

## PROCEDURES

```

//XTRDISK      PROC  A=A,CL=' ',CL1=' ',JOB LIB='FFS.JOPLIB',      00000100
//              LAB=BLP,RGN=60K,SAM=,USAM='(TAPE,DEFER)',          00000110
//              PTFJLBL='PTF.JOPLIB',                             00000200
//              VSAM=,XFDISP=SHR,XFNAME=,XFUNIT=2314,XFVCL=        00000300
//*****                                              00000400
//*                                                    * 00000500
//*      THIS PROC IS USED TO TRANSFER A DISK-RESIDENT INDEX DATA SET * 00000600
//* TO TAPE. THIS OPERATION CONDENSES THE INDEX DATA SET. THE TAPE * 00000700
//* SO CREATED IS A SEQUENTIAL DATA SET CONSISTING OF VARIABLE LENGTH * 00000800
//* BLOCKED RECORDS THAT CONTAIN THE SOURCE DATA AND CONTROL        * 00000900
//* INFORMATION FOR SUBSEQUENTLY RECONSTRUCTING THE SOURCE DATA.    * 00001000
//*                                                    * 00001100
//*      //STEPNAME EXEC XTRDISK,XFNAME=WW,SAM=XX,XFVCL=YY,VSAM=ZZ   * 00001200
//*                                                    * 00001300
//*      CHARLES W. HICKISCH MAJOR,USA PROJ CODE=763NIPS BRANCH=431  00001400
//*      DATE=MARCH 1, 1974                                           00001500
//*                                                    * 00001600
//*****                                              00001700
//*                                                    * 00001800
//XTR          EXEC PGM=UTNDXTR,PARM=DISK,REGION=&RGN              00001900
//STEPLIB DD    DISP=SHR,DSN=&PTFJLBL                             00002000
//          DD    DISP=SHR,DSN=&JOPLIB                             00002010
//INDEXDAM DD    DSN=&XFNAME,UNIT=&XFUNIT,VOL=&XFVCL,DISP=&XFDISP    00002100
//INDEXPRT DD    SYSOUT=(&A,&CL)                                   00002200
//INDEXSAM DD    DSN=&SAM,UNIT=&USAM,VOL=&VSAM,DISP=(NEW,KEEP),      *00002300
//          LABEL=(, &LAB)                                         00002400
//SYSPRINT DD    SYSOUT=(&A,&CL)                                   00002500
//SYSUDUMP DD    SYSOUT=(&A,&CL1)                                  00002600

```

```

//XRTAPE      PROC  A=A,CL=*,*,CL1=*,*,JOB LIB='FFS.JOB LIB',          00000100
//              LAB=RLP,NBRBLK=50,RCN=60K,SAM=,                        00000200
//              PTFJBL='PTF.JOB LIB',                                00000210
//              USAM='(TAPE9,,DEFF)',VSAM=,XFDISP=OLD,                C00000300
//              XTDISP='(NEW,KFEP)',XTNAME=,XTUNIT=2314,XTVOL=       00000400
//*****
//*              * 00000500
//*              * 00000600
//*      THIS PROC IS USED TO RECONSTRUCT A DISK-RESIDENT INDEX DATA * 00000700
//* SET FROM A PREVIOUSLY UNLOADED SEQUENTIAL VERSION OF THE INDEX * 00000800
//* DATA SET. * 00000900
//* * 00001000
//*      //STEPNAME EXEC XRTAPE,SAM=WW,XTNAME=XX,USAM=YY,XTVOL=ZZ * 00001100
//* * 00001200
//*      CHARLES W. HICKISCH MAJOR,USA PROJ CODE=763NIPS BRANCH=431 00001300
//*      DATE=MARCH 1,1974 00001400
//* * 00001500
//***** 00001600
//XTR          EXEC PGM=UTNDXTR,PARM=TAPE,REGION=ERGN 00001700
//STEPLIB DD   DISP=SHR,DSN=&PTFJBL 00001800
//          DD   DISP=SHR,DSN=&JOB LIB 00001810
//INDEXPRT DD   SYSOUT=(&A,&CL) 00001900
//INDEXSAM DD   DSN=&SAM,UNIT=&USAM,VOL=&VSAM,DISP=&XFDISP, C00002000
//          LABEL=(,&CLAB) 00002100
//SYSPRINT DD   SYSOUT=(&A,&CL) 00002200
//SYSUDUMP DD   SYSOUT=(&A,&CL1) 00002300
//XINDEX DD     DSN=&XTNAME,UNIT=&XTUNIT,VOL=&XTVOL, *00002400
//          DISP=&XTDISP,SPACE=(560,&NBRBLK), C00002500
//          DCB=(BLKSIZE=560,RECFM=F,KEYLEN=4,DSORG=DA) 00002600

```



DEFENSE COMMUNICATIONS AGENCY  
COMMAND AND CONTROL  
TECHINICAL CENTER  
WASHINGTON, D.C. 20301

IN REPLY  
REFER TO:

15 October 1977

TO DISTRIBUTION

SUBJECT: Change 3 to CSM UM 15-74, Job Preparation,  
Volume VIII, dated 15 October 1974

1. Insert the enclosed change pages and destroy the replaced pages according to applicable security regulations.
2. A list of Effective Pages to verify the accuracy of this manual is enclosed. This list should be inserted before the title page.
3. When this change has been posted, make an entry in the Record of Changes on the inside cover.

FOR THE DIRECTOR:

110 Enclosures  
Change 3 pages

A handwritten signature in cursive script, likely of J. Douglas Potter, is written over a circular stamp.

J. DOUGLAS POTTER  
Assistant to the Director  
for Administration

EFFECTIVE PAGES - 30 September 1977

This list is used to verify the accuracy of CSM UM 15-74, Volume VIII, after change 3 pages have been inserted. Original pages are indicated by the letter O, change 3 by the numeral 3.

<u>Page No.</u>	<u>Change No.</u>
Title Page	O
ii	O
iii-vi	3
vii	O
1-2	3
3-14	O
15-24	3
25-38	O
39-40	3
41-42	O
43-44	3
45-50	O
51-54	3
54.1-60	O
61	3
61.1-61.16	3
62-66.2	3
67-72	3
73-74	O
75-76	3
76.1-76.2	3
77-81	3
81.1	3
82-84	3
85-86	O
87-88	3
89-92	O
93-106	3
106.1-106.2	3
107-108	O
109-112	3
113-114	O



EFFECTIVE PAGES - (continued)

<u>Page No.</u>	<u>Change No.</u>
115-118	3
119-122	0
123-130	3
131-142	0
143-144	3
145-146	0
147-148	3
148.1-148.2	3
149-150	0
151-152	3
153-162	0

# CONTENTS

Section		Page
	ACKNOWLEDGMENT.....	ii
	ABSTRACT.....	vii
#1	INTRODUCTION.....	1
2	FEATURES OF NIPS 360 PFS PROCEDURES.....	2
2.1	Symbolic Parameters.....	2
2.2	File Name Conventions.....	3
2.3	File Block Size Conventions.....	3
2.3.1	Default File Block Size.....	4
2.3.2	User Specified File Block Size.....	4
2.4	Example of File Name Conventions and Symbolic Parameters.....	4
2.4.1	Single Data Base.....	4
2.4.2	Multiple Data Bases.....	6
2.4.3	File Libraries.....	8
2.5	Catalog Requirements.....	8
2.6	Checkpoint/Restart.....	9
2.6.1	Checklist for Using Checkpoint/Restart.....	11
2.6.2	Sample Job Setup.....	11
2.7	Segmented Files.....	13
2.8	Source Language Storage.....	13
3	COMPONENT PROCEDURE DESCRIPTIONS.....	14
#3.1	PM Procedure (XPM).....	14
3.1.1	Sample Job Setup.....	18
3.2	FR Procedure (XFR).....	27
3.2.1	Sample Job Setup.....	27
3.2.2	File Structure, Revision, and Maintenance....	28
3.3	PS Procedure (XPS).....	28
3.3.1	Sample Job Setup.....	28
3.4	ISAM to SAM Procedure (XISTOS).....	29
3.4.1	Sample Job setup.....	30
3.5	OP Procedures (XOP, XOPSD).....	30
3.5.1	XOPSD Sample Job for Single File Processing..	32
3.5.2	XOPSD Sample Job for Merge File Processing...	32
3.5.3	XOP Sample Job.....	33
3.5.4	Segmented File Sample.....	34
3.5.5	Output Data Set Overrides.....	35
3.5.6	Additional DD Card for Tape Output.....	34
3.6	QRT/QDF Utility Procedure (XQRTQDF).....	36
3.6.1	Sample Job Setup.....	36
3.7	QUIP Procedures (XQUIP, XQUIPSD).....	36
3.7.1	Sample Job Setup.....	37
3.7.2	Interfile Output (IFO).....	39

Section		Page
#3.8	RASP Procedure (XRASP).....	39
3.8.1	Sample Job Setup.....	40
#3.9	SAM to ISAM Procedure (XSTOIS).....	43
3.9.1	Sample Job Setup.....	44
3.10	Subroutine Loader Procedure (XSUBIDR).....	44
3.10.1	Sample Job Setup.....	44
3.11	Table Generation Procedure (XTABGEN).....	45
3.11.1	Sample Job Setup.....	45
3.12	Source Language Library Procedure (XUTSOURC).....	45
3.12.1	Sample Job Setup.....	45
3.13	1410 to 360 Conversion (X360CON).....	46
3.13.1	Sample Job Setup.....	46
3.14	360 to 1410 Conversion (X1410CON).....	47
3.14.1	Sample Job Setup.....	47
3.15	List Logic Statement/Report Names Procedure (XDMELIB).....	47
3.15.1	Sample Job Setup.....	48
3.16	Subroutine Check Procedure (XSUBCHK).....	48
3.16.1	Sample Job Setup.....	48
3.17	Classification Change Procedure (XCLASS).....	48
3.17.1	Sample Job Setup.....	49
3.18	Data Field Scan Procedure (XUTFSCAN).....	50
3.18.1	Sample Job Setup.....	50
3.19	Index Specification Procedure (XSP).....	51
3.19.1	Sample Job Setup.....	51
3.20	Unload Index Data Set Procedure (XTRDISK).....	52
#3.20.1	Sample Job Setup.....	52
3.21	Load Index Data Set Procedure (XTRTAPE).....	53
#3.21.1	Sample Job Setup.....	53
3.22	Keyword Analysis (XKA).....	53
3.22.1	Sample Job Setup.....	53
3.23	Dictionary Maintenance Utility (XKM).....	54
3.23.1	Sample Job Setup.....	54
3.24	Format Definition Translator Procedure (XUTODE).....	54
3.24.1	Sample Job Setup.....	54.1
4	SUPPLEMENTARY PROCEDURES DESCRIPTIONS.....	55
4.1	Restore Answers Procedure (XRESTANS).....	55
4.1.1	Sample Job Setup.....	55
4.2	Restore Library Procedure (XRESTLIB).....	56
4.2.1	Sample Job Setup.....	56
4.3	Save Answers Procedure (XSAVEANS).....	57
4.3.1	Sample Job Setup.....	57
4.4	Save Library Procedure (XSAVELIB).....	57
4.4.1	Sample Job Setup.....	58
5	PRODUCTION PROCEDURES DESCRIPTION.....	59
5.1	PM Production Procedure (XPMEX).....	59
5.2	OP Production Procedure (XCPEX, XOPSDEX).....	59

Section		Page
5.3	Compression and Compaction of Data Records...	60
5.4	RASP Production Procedure (XRASPEX) .....	61
#6	S/370 VSAM CONSIDERATIONS .....	61.1
#6.1	VSAM Service Routine IDCAMS .....	61.1
#6.1.1	Creating a NIPS User Catalog .....	61.1
#6.1.2	Defining a Cluster for a VSAM File .....	61.2
#6.1.3	Converting an ISAM File to VSAM .....	61.3
#6.1.4	Deleting a Cluster .....	61.4
#6.2	File Maintenance .....	61.4
#6.2.1	VSAM Generate .....	61.4
#6.2.2	VSAM Update .....	61.5
#6.2.3	VSAM Transactions .....	61.5
#6.2.4	Processing SAM Files with a VSAM FFT/LS .....	61.6
#6.2.5	Other FM VSAM Considerations .....	61.7
#6.3	File Revision .....	61.8
#6.4	File Structure .....	61.9
#6.5	VSAM to SAM Procedure (XISTOS) .....	61.9
#6.6	OP .....	61.10
#6.7	QUIP .....	61.11
#6.8	RASP .....	61.12
#6.9	SAM to VSAM Procedure (XISTOIS) .....	61.13
#6.10	XDMPLIB .....	61.13
#6.11	XCLASS .....	61.13
#6.12	XUTIFSCAN .....	61.14
#6.13	XSP .....	61.14
#6.14	XTRDISK .....	61.15
#6.15	XKA .....	61.15

#### APPENDIX

#	A	SYMBOLIC PARAMETER DEFINITIONS.....	62
	B	INPUT SOURCE DDNAME.....	71
#	C	PROCEDURES DESCRIPTIONS.....	71
	D	NIPS PROCEDURE RETURN CODES.....	75
#	E	NIPS PROCEDURE DD STATEMENT USAGE.....	77
#	F	PROCEDURE LISTINGS.....	108

XCLASS  
 XDMPLIB  
 XFM  
 XFMEX  
 XFR  
 XFS  
 XISTOS

XKA  
XKM  
XOP  
XCPEX  
XOPSD  
XCPSDEX  
XQRTQDF  
XQUIP  
XQUIPSD  
XFASP  
XRASPEX  
XRESTANS  
XRESTLIB  
XSAVFANS  
XSAVELIB  
XSP  
XSTOIS  
XSUBCHK  
XSUBLDF  
XTABGEN  
XTP  
XTRDISK  
XTRTAPE  
XUTFSCAN  
XUTSOURC  
X1410CCN  
X360CCN

DISTRIBUTION..... 153



## JOB PREPARATION

### Section 1

#### INTRODUCTION

This volume is intended to familiarize the user with the features available using NIPS 360 FFS Job Preparation procedures and the manner in which they are used.

Section 2 describes symbolic parameters and file naming conventions. It also illustrates general approaches to running jobs using single or multiple data bases and file libraries.

Section 3 contains JCL examples for each component and additional information pertinent to each procedure.

Section 4 contains JCL examples for S/360 Utilities which are used to dump and restore RASP answers and the User File Library from disk to tape and vice versa.

Section 5 describes procedures which are used in File Maintenance, Output Processing and the Retrieval and Sort Processor component.

\* Section 6 describes the procedures and considerations necessary to create, maintain and process NIPS data bases using the S/370 Virtual Storage Access Method (VSAM).

## JOB PREPARATION

### Section 2

#### FEATURES OF NIPS 360 PPS PROCEDURES

NIPS 360 PPS has single-step cataloged procedures using symbolic parameters. This feature allows a user to run all PPS jobs with no DD statement overrides unless one of the following conditions exist:

- a. PASP CP, or QUIP run with more than three data bases and/or more than three file libraries
- b. PM run with transaction source from tape or disk
- c. PM run with auxiliary output on tape or disk
- d. QUIP run storing a query into a user library.
- e. OP run with record output.

The procedures provide a convention for naming user data files and file libraries which will be discussed in subsequent paragraphs.

#### 2.1 Symbolic Parameters

Symbolic parameters simplify the overriding of DD statements in the execution of jobs. Basically, they allow the user to equate names, units, and volumes to symbolic parameters in the EXEC card without concern for the step name or the order of DD cards within a procedure. The use of symbolic parameters does not preclude the overriding of DD cards. A DD card override takes precedence over symbolic parameters. Each procedure has default options for every symbolic parameter not referenced by the user in his run. See examples in section 2.3.

## JOB PREPARATION

If the input is a SAM file on unlabeled magnetic tape and the file block size is not 1,004, use symbolic parameter BSIZE to indicate the input file's block size.

In some cases, the user may need to provide JCL information that is not included in the procedure. Overriding DD cards must have a DD statement name which includes the stepname (FM) and the name of the DD statement to which the override applies. All override DD cards must be placed in the order that they appear in the procedure.

If the transaction source is a single tape or disk file, the user must include an override DD statement named FM.TRANS which specifies the DSNNAME, DISP, UNIT, and VOLUME parameters plus the DCB parameters if the transaction file is an unlabeled tape. When multiple transaction sources are to be used in an FM execution, the user must provide a DD statement for each source. The DD statement must be named PSTRANxx for sequential tape or disk transactions and ISTRANxx for ISAM NIFS files. The XX may be a unique user specified ID for each DD statement. Parameters to be specified in each of the added transaction DD statements are the same as those required in the FM.TRANS DD statement. The FM.TRANS override DD must not be used when utilizing the multiple transaction source capability as it is used for describing single source transactions.

The following DD statements would be required only if the user is producing auxiliary output files by using the WRT, WT2, WT3, WT4 or WT5 instructions in his logic statement. These statements must include the DSNNAME parameter and should include UNIT, VOLUME and DISP parameters. If the user specifies a direct access device for output, he must also include a space parameter. DCB parameters should be specified if other than standard parameters are desired.

A DD statement named FM.AUX1, is used to identify the name given to the first auxiliary output file. This file is produced by the POOI instruction WRT.

A DD statement, FM.AUX2, is used to identify the name given to the second auxiliary output file. This file is produced by the POOI instruction WT2.

## JOB PREPARATION

A DD statement, FM.AUX3, is used to identify the name given to the third auxiliary output file. This file is produced by the POOL instruction WT3.

A DD statement, FM.AUX4, is used to identify the name given to the fourth auxiliary output file. This file is produced by the POOL instruction WT4.

A DD statement FM.AUX5, is used to identify the name given to the fifth auxiliary output file. This file is produced by the POOL instruction WT5.

A DD statement must always be used to identify the user's input deck. The format of this card is (//FM.SYSIN DD \*). This is followed by the user's input deck, followed by a /\* card to indicate the end of the input deck.

Examples of typical FM run decks follow.

The FM component is designed to use a disk sort provided enough disk sort work space (DD cards SORTWK01-06) is available. The FM component required this space to sort the transactions prior to their being added to the data file. As these transactions are processed by FM prior to sorting, FM calculates the amount of disk space required for the sort, and if this space is not available it will default to a tape sort. When this occurs the following DD cards must be added to the standard procedure:

```
//TAPEWK01 DD parameters defining a 9-track tape unit.  
//TAPEWK02 DD parameters defining a 9-track tape unit.  
//TAPEWK03 DD parameters defining a 9-track tape unit.  
//TAPEWK04 DD parameters defining a 9-track tape unit.
```

# In addition, if the number of transactions necessitates that SORTIN and SORTOUT will contain more than one tape volume, the following JCL changes are needed. Otherwise, only the transactions on the last volume specified on your VOLUME parameter will be used to update the file.

```
//FM.SORTIN DD UNIT=(2400,,DEFEF),VOL=SER=(.....)  
//FM.SORTOUT DD UNIT=APP=SORTIN,VOL=SER=(some tapes as SORTIN)
```

## JOB PREPARATION

The above DD cards are not distributed with the standard procedure. The space defined for the disk sort work area is adequate for most applications.

When generating an ISAM file, space allocation in terms of INDEX, PRIME, and CVFLOW areas must be specified for the new file. This is done utilizing the INDEX, PRIME, and OVFLOW symbolic parameters. These parameters default to 1, 5, and 1 cylinders respectively. For example, if 1 cylinder of index, 20 cylinders of prime area and 5 cylinders of independent overflow area are desired when generating the TEST360 file, the following EXEC card would be coded:

```
//FMGEN EXEC XFM,GEN=,ISAM=TEST360,PRIME=20,CVFICW=5
```

The XFM procedure will also maintain SAM data files. The use of the SAM or ISAM parameter specifies the type of input file to File Maintenance. The type of file must always agree with that specified on the FM control card, with one exception. A SAM file may be generated from an ISAM FFT. For this one exception, the ISAM symbolic parameter would be used to specify the input FFT and the FM control card would specify 'TAPE'.

When updating a SAM file, the updated file will normally be written on the data set (tape) specified by the FMSAMOUT DD statement. However, if any record controls are being changed by use of the MCT or MCW POOL instructions, the updated file will be written on the data set (tape) specified by the FMNDATA DD statement. These data sets are normally written on a nine-track tape. A message is written on the console stating the file name and which tape is to be saved at the completion of FM.

The XFM procedure is generalized since it has a variety of applications: SAM or ISAM data files in either the generate or update mode. To avoid possible run terminations due to insufficient space for the NEWFILE DD statement, special provisions have been made to DUMMY this statement in the XFM procedure. Also, this statement must be DUMMY when updating a multi-volume ISAM data file. No space will be allocated for this statement unless GEN= is coded on the EXEC card. Coding of GEN= on the EXEC card is applicable for an ISAM generate run only. Special consideration has been directed to the FMSAMOUT and FMNDATA DD statements.



## JOB PREPARATION

These DD statements have been put to DUMMY. They should remain in the DUMMY status for all ISAM runs. For all SAM runs, the user must code SAMOUT= on the execute card to allow allocation of the FMSAMOUT and FMNDATA data sets (tapes).

During COM and LIB mode runs, if the FFT and logic statements are on a sequential file, no library action will be performed. However, the XFM procedure may be used for debugging a logic statement. If the FFT is on an indexed sequential file, library action may be performed and logic statements may be added to the library.

In FM the default for the processing block size is 16,000 bytes. The user may override this size by using the PARM='PFSIZE=nK' or PARM='PB=nK' parameter on the EXEC card, where n may be any integer from 1 to 99 inclusive.

### 3.1.1 Sample Job Setup

The following examples generally use the ISAM symbolic parameters. Except where noted, these same examples can be used for SAM files by using the SAM, VSAM and USAM parameters in place of the respective ISAM parameters. The FM control card must also be changed to TAPE instead of DISK file update. For SAM files, SAMOUT= must be coded on the EXEC card.

- a. Compile Logic Statements Only - The following FM run deck would be used to compile logic statements for the TEST360 file. The purpose of this run is to produce listings of the logic statement source cards for use in debugging new logic statements. This type of run would be used as the first step in setting up a new file. The file is not cataloged but resides on a 2314 disk pack.

A typical run deck follows:

```
// EXEC      XFM,ISAM=TEST360,VISAM='SER=MYPACK'
//FM.SYSIN DD *
$FMS/COM,TEST360
.
.
.
. LOGIC STATEMENT LIBRARY UPDATE DECK
.
```

JOB PREPARATION

.  
.  
/\*

- b. Update Logic Statement Library - The following run deck could be used to update the Logic Statement Library for the TEST360 file which is cataloged.

```
// EXEC      XFM,ISAM=TEST360
//FM.SYSIN    DD      *
$FMS/LIB,TEST360
.
.   LOGIC STATEMENT LIBRARY UPDATE
.
/*
```

If the file is a SAM file, logic statements can only be added during a GEN or UPD run.

- c. Update a File (Card Transactions) - The following deck could be used to update the TEST360 file using card transactions. The first transaction report name is ONE. The second report name is TWO. The user's subroutine library is named TEST360L and is cataloged. The processing block size will be increased from 16,000 bytes to 20,000 bytes.

```
//          EXEC XFM,ISAM=TEST360,LIB=TEST360,
//          PARM='PFSIZE=20K'
//FM.SYSIN    DD      *
$FMS/UPD,TEST360,ONE
.
.   TRANSACTIONS FROM REPORT ONE
.
.   NEW REPORT TWO
.
.   TRANSACTIONS FROM REPORT TWO
.
/*
```

- d. Update a File (Tape Transactions) - The following deck could be used to update the TEST360 file with tape transactions. The transaction report name is XYZ. The transaction file label is TAPXACT and it

# JOB PREPARATION

resides on tape with the volume label 1234. The installation has 2400 tape drives, and the transaction tape has standard labels.

```
// EXEC      XFM,ISAM=TEST360,LIB=TEST360
//PM.TRANS DD  DSN=TAPEX,UNIT=2400,          X
//          VOLUME=SER=1234,DISP=CLD
//PM.SYSIN DD  *
$FMS/UPD,TEST360,XYZ,,DISK,TAPE
/*
```

- e. Compile Logic Statements, Update a File (Tape Transactions), and Produce Two Tape Auxiliary Outputs - The following run deck could be used to compile logic statements and update the file TEST360 with a tape transaction file labeled TAPEX. The transaction tape volume label is 2345. The transaction report name is ABC; 2400 series tape drives are available for mounting tapes. Two auxiliary output tapes are produced, named AUDITA and AUDITB. These are to be produced on tapes with serial numbers TAPEA and TAPEB.

```
// EXEC      XFM,ISAM=TEST360,LIB=TEST360
//PM.TRANS DD  DSN=TAPEX,UNIT=2400,          X
//          VOLUME=SER=2345,DISP=CLD
//PM.AUX1 DD  DSN=AUDITA,UNIT=2400,          X
//          VOLUME=SER=TAPEA,DISP=(,KEEP)
//PM.AUX2 DD  DSN=AUDITB,UNIT=2400,          X
//          VOLUME=SER=TAPEB,DISP=(,KEEP)
//PM.SYSIN DD  *
$FMS/UPD,TEST360,ABC,IS,DISK,TAPE
.
.
. LOGIC STATEMENT LIBRARY UPDATE DECK
.
.
/*
```

- f. Compile Logic Statement and Generate a File (Card Transactions) - The following run deck could be used to compile logic statements for a TEST360 file and to generate the TEST360 file using card transactions. The transaction report names are A and B. The users subroutine library is named

# JOB PREPARATION

TEST36CL. The index, prime and overflow areas for the generated file are 1, 25 and 10 cylinders, respectively.

```
// EXEC      XFM,ISAM=TEST360,LIE=TEST360,      Y
//          PRIME=25,CVFLOW=10,GEN=,INDEX=1
//FM.SYSIN    DD      *
$PMS/GEN,TEST360,,LS,DISK
.
.   LOGIC STATEMENT LIBRARY UPDATE DECK      .
.
.   NEW REPORT A                             .
.
.   REPORT A TRANSACTIONS                    .
.
.   NEW REPORT B                             .
.
.   REPORT B TRANSACTIONS                    .
.
/*
```

If a SAM file were being generated, the PRIME, INDEX and OVFLOW parameters would not be used.

9. Generate an ISAM File (Card Transactions) Using the FFT and Logic Statement Library of Another File - The following run deck could be used to generate a file having the same FFT and Logic Statement Library as an existing file. (The data base of the new file must still be supplied via transaction inputs.) It is assumed that the input data file, TEST360, is cataloged and that the required logic statements have been added to the Logic Statement Library. The new file will be named MYFILE, and will be generated with card transactions. At the end of the FM run, both TEST360 and MYFILE will be cataloged data sets. The transaction report name is REPORT. The user's subroutine library is named SUESI and is cataloged.

```
// EXEC      XFM,ISAM=TEST360,LIE=SUES,GEN=
//FM.SYSIN    DD      *
$PMS/GEN,MYFILE,REPORT
```

# JOB PREPARATION

```
.
.
. TRANSACTIONS FOR EXECUT
.
.
/*
```

- h. Generate a SAM File (ISAM FFT) - For GEN mode sequential processing, the input file containing the FFT (and logic statements, if desired) may be either a sequential or an indexed sequential file. The following is an example of using card transactions to generate a SAM file from an ISAM file consisting of FFT and logic statements. The TEST360 file is cataloged. The sequential data file will be cataloged at the end of the processing with the name specified on the FMS control card padded with a suffix character of 'S'.

```
// EXEC XFM,ISAM=TEST360,SAMOUT=
//FM.SYSIN DD *
$FMS/GEN,TEST360,RPT,,TAPE,CARD
.
.
. TRANSACTIONS FOR REPORT RPT
.
/*
```

Note that GEN= is not coded on the EXEC card for a SAM generate run. It only applies to an ISAM generate run. Note also that SAMOUT= must be coded on the EXEC card for any FM run that produces a SAM output file. The Indexed sequential data file will be cataloged at the end of the processing. If a qualified data set name was used, and the last segment of the name matches the name on the FMS control card, the fully qualified name will be used as the catalog entry.

1. Update a SAM File and Add Logic Statements - For UPD mode sequential processing, the input file must be a sequential file and SAMOUT= must be coded on the EXEC card.

```
// EXEC XFM,SAM=TEST360,SAMOUT=
```



## JOB PREPARATION

```
//FM.SYSIN DD *
$FMS/UPD,TEST36C,RPT,LS,TAPE,CARD
.
. LOGIC STATEMENTS
. TRANSACTIONS FOR EXECUT RPT
.
.
/*
```

Note: Although the sequential file is cataloged with a suffix character 'S', it should be specified without the suffix on the FMS control card and the execute card.

- j. Generate a First Segment From an FFT - The following JCL and input control cards would be used to generate an initial segment from an ISAM file consisting of FFT and logic statements.

```
// EXEC XFM,ISAM=TEST36C,SAMCUT=,
// LAE=SL
//FM.SYSIN DD *
$FMS/GEN,TEST36C,RPT,,TAPE,CARD,SEG
$SEG M00001 M00999
.
.
. CARD TRANSACTION FOR REPORT RPT
.
.
/*
```

- k. Generate a Segment From Another Segment - The following JCL and input control cards would be used to generate a segment. (The first segment was created using the JCL and control cards in sample j above.)

# JOB PREPARATION

```
// EXEC   XFM,SAM=TEST360,SAMOUT=,
//          LAB=SL
//FM.SYSIN DD *
$FMS/GEN,TEST360,RPT,,TAPE,CARD
$SEG   N00001   N00999
.
.
.   CARD TRANSACTIONS FOR REPORT RPT
.
.
/*
```

1. Generate a New Segment From a Segment and Add a New Segment Record - The following JCL and input control cards would be used to generate a new segment from an existing segment and add new segment control records.

```
// EXEC   SAM=TEST360,SAMOUT=,LAB=SL
//FM.SYSIN DD *
$FMS/GEN,TEST360,RPT,,TAPE,CARD
$ADD      JC0001   JC0999           N36098
$SEG      W00001   W00999
.
.
.   CARD TRANSACTIONS
.
.
/*
```

Note: After generation of a segment using the FFT and logic statements from the previous segment, the new segment will contain the FFT and logic statements and old segment control records from the previous segments. A new segment control record for the new segment and the new data records generated by the transactions will also be on the new segment.

## JOB PREPARATION

### QUIP SOURCE STATEMENTS

/\*

#### 3.7.2 Interfile Output (IFO)

Either the XQUIF or XQUIPSD procedures may be used with IFO. When XQUIPSD is used, the primary file may be SAM or ISAM and is specified by the SAM or ISAM symbolic parameter. When XQUIF is used, the primary file is the RASP answer set and is specified in the same manner as for a single file query.

With either procedure, up to nine secondary files may be referenced. All secondary files must be ISAM. The symbolic parameters, ISAM1 and ISAM2, are provided for two secondary files.

If the input is a SAM file on unlabeled magnetic tape and its block size is not 1,004 bytes, its block size must be specified with the BSZFILE symbolic parameter.

#### 3.8 RASP Procedure (XRASP)

This procedure may be used for either single-file or merge-file retrievals. If the retrieval is to be added to the file library identified by the LIB parameter, LIBDISP=OLD must be specified on the EXEC card. Use care when specifying file libraries using LIB, LIB1, LIB2 parameters since the procedure will suffix the specified name with an 'L'. Indexed files will be suffixed with an 'X' when the parameters XINDEX, INDEX1, and INDEX2 are specified, while SAM files will be suffixed with an 'S' when using the SAM, SAM1, and SAM2 parameters.

If an assembler list of the retrieval is desired, code PARM='LIST' on the EXEC card.

# PARM='NOFL' on the EXEC card causes the FFT and logic statements not to be copied to the QDF. This option should be used only when the QDF is input to OP.

## JOB PREPARATION

# If the file being queried is indexed, the File Indexing mode of retrieval will automatically be invoked if the retrieval logic allows. The user may override the Secondary Indexing mode of retrieval by using the PARM='INDEX=NO' parameter on the EXEC card. However, the presence of a KEYWORD statement in the retrieval negates the PARM override so that indexing will be involved as long as it is feasible.

### 3.8.1 Sample Job Setup

The following JCL could be used to retrieve from TEST360 using the file library TEST360L. Both files are cataloged.

```
//RET      EXEC      XRASP,ISAM=TEST360,LIB=TEST360
//PASP.SYSIN DD      *
```

RETRIEVAL DECK

/\*

The following JCL could be used if two libraries are required and TEST360 is on tape (9 track). The TEST360S data base and both libraries (TEST360L and TRAINEFL) are cataloged. The retrieval is to be added to the TESTERL library.

```
//RET      EXEC      XRASP,SAM=TEST360,LIB=TEST360,      X
//          LIB1=TRAINER,LIBDISP=OLD
//PASP.SYSIN DD      *
```

RETRIEVAL DECK

/\*

The XRASP procedure can be used to query a segmented data file. If only one segment is to be queried, the deck setup is the same as for a SAM FILE execution. If more than one segment is to be queried, additional DD cards must be provided to define the additional segments. The first segment may be defined through the symbolic parameters, but the additional DD statements must completely define the data set.

## JCB PREPARATION

be performed with library TEST360L. TRAINERS is an indexed file.

```
//RETI EXEC XRASP,ISAM=TEST360,SAM=TRAINER,LIB=TEST360, X
//          LIB1=TRAINER,LIBDISP=OLD,XINDEX=TRAINER
//RASP.SYSIN DD *
//          (RASP CCNTECL AND SOURCE LANGUAGE STATEMENTS)
/*
```

### 3.9 SAM TO ISAM Procedure (XSTCIS)

This procedure is used to load a tape data file (SAM) to disk (ISAM). Space is allocated separately for INDEX, PRIME, and OVFLOW areas. The procedure defaults these areas to 1, 40, and 5 cylinders, respectively. The disposition of the new disk file defaults to KEEP.

If the input is SAM file on unlabeled magnetic tape and its block size is not 1,004 bytes, its block size must be specified with the BSZFILE symbolic parameter.

The BSZNEW symbolic parameter is used to specify a new block size for the ISAM file. If it is not used, the ISAM block size will be the same as the input SAM block size.

# The procedure also can be used to create an ISAM file in the compression/compaction form or to re-create a standard form ISAM file. The PARM parameter can be used for specifying the option. The default value is the form of the input file. The options are:

PARM=COMPRESS	for compression only
PARM=COMPACT	for compaction only
PARM='COMPRESS,COMPACT'	for both compression and compaction
PARM=EXPAND	for reversing the compression/compaction process to produce standard form data records



## JOB PREPARATION

All unused PRIME space will be filled with system generated PAD records. The PARM option NCPAD may be used if this feature is not desired, e.g.:

```
PARM=NCPAD
```

### 3.9.1 Sample Job Setup

The following JCL could be used to load the TEST360S file from tape-to-disk. It is desired to increase the PRIME area to 50 cylinders and catalog the new data set. The SAM file is not cataloged.

```
//STOIS EXEC XSTCIS,SAM=TEST360,PRIME=50, X
//          VSAM='SER=MYTAPE',ISAM=TEST360, X
//          VISAM='SER=MYPACK',NDISP=CATLG
/*
```

### 3.10 Subroutine Loader Procedure (XSUBLDR)

This procedure is used to place user subroutines on the file library. The two inputs are the user subroutine in load module form on a temporary or permanent data set and a control card. The MODLIB parameter is used to specify the name of the data set containing the input subroutine. It defaults to &TEMP which is the name of the data set containing the subroutine if the ASMFCL procedure has been used in a prior step of the job to assemble and link edit the subroutine.

#### 3.10.1 Sample Job Setup

The following JCL could be used to assemble, link-edit, and add a subroutine to the TEST360L library which is cataloged. The subroutine is named DTGIS.

Note: The MODLIB DSNAME is not specified on the EXEC card since it defaults to the DSNAME of the assembler job step output (&TEMP).

```
//ASML EXEC ASMFCL
//AS*.SYSIN DD *
```

SUBROUTINE SOURCE DECK (DTGIS)

```
/*
```

## JOB PREPARATION

### 3.19 Index Specification Utility (XSP)

This procedure is used to specify Keyword Indexing information for a data file without running a File Structure or File Maintenance job. The disk-resident Index Data Set may be either generated or updated from either an ISAM or SAM data file.

Both SUE/TAB and INDEX statements are accepted and must be specified through the SYSIN device. Both additions and deletions of indexed fields may be made in the same run.

If, for any reason, the Index Data Set is destroyed but the file remains operational, the user has the option to recreate the Index Data Set to have it correspond with the existing file by specifying PARM='GEN' and XDISF=NEW. If the input is a SAM file on unlabeled magnetic tape and its block size is not 1,004 bytes, its block size must be specified with the ISZFILE symbolic parameter.

#### 3.19.1 Sample Job Setup

The first example is used to generate a new Index Data Set from an existing ISAM data file.

```
//GENXSP EXEC XSP,XINDEX=TEST360,XLISP=NEW,
//          XVOL='SER=MYPACK',LIB=TEST360,
//          VIIB='SER=MYPACK',ISAM=TEST360,
//          VISAM='SER=MYPACK',
//XSP.SYSIN DD
//          INDEX STATEMENTS
/*
```

The second example is used to update an existing Index Data Set from an existing SAM data file.

```
//UPDXSP EXEC XSP,XINDEX=TEST360,XVOL='SER=MYPACK',
//          LIB=TEST360,VLIB='SER=MYPACK',
//          SAM=TEST360,VSAM='SER=MYTAPE',
//          LAB=SL,DEN=2,
//          SAMOUT=,VSMOUT='SER=NEWTAP',
//XSP.SYSIN DD
//          INDEX STATEMENTS
/*
```

## JOB PREPARATION

The third example is used to recreate an Index Data Set from an existing SAM file containing Index Descriptor records. In this example no changes are desired to the indexed fields. However, the Index Descriptor records may be updated by including the proper Index Statements.

```
//RECXSP EXEC XSP,PARM='GEN',XINDEX=TEST360,  
//          XVOL='SER=MYPACK',IIB=TEST360,  
//          VLIB='SER=MYPACK',SAM=TEST360,  
//          VSAM='SER=MYTAPE',LAB=SL,  
//          DEN=2,SAMCUT=,VSMOUNT='SER=NEWTAP',  
//          XDISP=NEW  
//XSP.SYSIN DD  
/*
```

### 3.20 Unload Index Data Set Procedure (XTRDISK)

This procedure is used to copy a disk-resident, direct-access organization, Index Data Set to a tape-resident, sequential-access organization data set. The XFNAME and XFVOL symbolic parameters must be supplied for the Input Index Data Set. The XTNAME and XTVOL symbolic parameters must be supplied for the unloaded data set residing on tape. No control statements are required.

# If statistics on the content of the index data set are desired, STAT=YES should be designated. When this option is chosen the corresponding data file name and volume must also be specified. The parameters are ISAM and VISAM for an ISAM data base and SAM and VSAM for a SAM data base.

#### 3.20.1 Sample Job Setup

# The following JCL would unload the TEST360X Index Data Set from a 2314 disk pack to a nine-track labeled tape.

```
//STEPNAME EXEC XTRDISK,  
//          XTNAME=INDEXSAM,XTVCI='SER=MYTAPE',  
//          XFNAME=TEST360X,XFVOL='SER=MYPACK'  
/*
```

The following JCL would unload and print the contents of the TEST360X index data set.

## JOB PREPARATION

```
//STEPNAME EXEC XTRDISK,STAT=YES
*//          XTNAME=INDEXSAM,XTVOL='SER=MYTAPE',
//          XFNAME=TEST36(X,XFVOL='SER=MYPACK',
/*
```

### 3.21 Load Index Data Set Procedure (XTRTAPE)

This procedure is used to reconstruct a disk-resident direct access organization Index Data Set from a previously unloaded sequential access organization data set. The XFNAME and XFVOL symbolic parameters must be supplied for the input tape. The XTNAME and XTVOL symbolic parameters must be supplied for the Output Index Data Set on disk.

#### 3.21.1 Sample Job Setup

\* The following JCL would load the TEST360X Index Data Set to a 2314 disk pack from a nine-track labeled tape.

```
//STEPNAME EXEC XTRTAPE,
//          XTNAME=TEST360X,XTVOL='SER=MYPACK',
*//          XFNAME=INDEXSAM,XFVOL='SER=MYTAPE'
/*
```

### 3.22 Keyword Analysis (XKA)

This procedure is used to obtain printed listing of text, non-keywords, and keywords that occur in a data base without updating the index data set. No DD card overrides are necessary, but a DD statement named SYSIN must always be included to identify the user's input. A file name must be specified using either the ISAM or SAM symbolic parameter. If a user scan subroutine, stop word table, or dictionary is required, the library name must be specified with the LIB symbolic parameter.

#### 3.22.1 Sample Job Setup

The following JCL would be used to analyze keyword fields in the TEST360 ISAM data base:

## JCB PREPARATION

```
//STEPNAME EXEC XKA,ISAM=TEST360,LIB=TEST360,  
//          VISAM='SER=MYPACK',VIIB='SEF=MYPACK'  
//XKA.SYSIN DD *  
          (user supplied control statements)  
/*
```

### 3.23 Dictionary Maintenance Utility (XKM)

This procedure is used to generate or update keywords, stop word tables and dictionaries. More than one stop word table or dictionary may be created or updated in one run provided that all are members in one library (specified by the LIB parameter), stop word table and dictionary name specifications, maintenance commands, and data are accepted through the SYSIN device.

#### 3.23.1 Sample Job Setup

The following JCL would be used to maintain any or all stop word tables and dictionaries in the TEST360 library.

```
// EXEC XKM,LIB=TEST360  
//SKM.SYSIN DD *  
      Table or Dictionary Names  
      Maintenance Commands  
      Data (keyword specs)  
/*
```

### 3.24 Format Definition Translator Procedure (XUTODE)

This procedure is used to place format definitions on the user library. The input definition statements may be in punched cards or in card image records stored in a library. More than one format definition may be added to the library at a time.

#### 3.24.1 Sample Job Setup

The following JCL would be used to add a format definition in punched cards to the user library, TEST360.



## JOB PREPARATION

### 5.4 RASP Production Procedure (XRASPEX)

The procedure XRASPEX is used to execute previously compiled single-file or merge-file retrievals. It may not be used to compile a retrieval or to add a retrieval to a library. The intended use of procedure XRASPEX is for production retrievals only.

Specifications for symbolic parameters and override DD statements for the production procedure XRASPEX are the same.

## JOB PREPARATION

### #Section 6

#### S/370 VSAM CONSIDERATIONS

This section addresses the procedures and considerations under which the S/370 user of NIPS FFS may create and process a NIPS file using the Virtual Storage Access Method (VSAM). Each NIPS procedure which might be used to process a VSAM file is examined and the VSAM service routine IDCAMS is examined for use with NIPS VSAM files.

The sample jobs will omit the job card which is always required and standard for each installation.

NOTE: NIPS FFS will not currently process a VSAM NIPS file in the terminal environment. Also NIPS FFS will not process VSAM files which are not NIPS files.

#### 6.1 VSAM Service Routine IDCAMS

VSAM processing requires that VSAM files be defined before they can be accessed by any program or system, including NIPS. The VSAM service routine IDCAMS is used for this purpose. By making use of the various options available, the user can create user catalogs, define clusters (allocate and catalog data sets), copy data sets, move data sets from one operating system to another, and recover from certain types of data damage.

Four of the IDCAMS options are of interest to the NIPS user.

- o Creating a NIPS user catalog
- o Defining a cluster for a VSAM file
- o Converting an ISAM FILE to VSAM
- o Deleting an unneeded cluster

##### 6.1.1 Creating a NIPS User Catalog

All VSAM data sets must be cataloged on a user or a master catalog. In normal applications, the user's VSAM file will be cataloged on the NIPS user catalog. This provides for user control of his data sets. The following JCL could be used to create a NIPS user catalog:

UNCLASSIFIED

NATIONAL MILITARY COMMAND SYSTEM SUPPORT CENTER WASH--ETC F/G 9/2  
NMCS INFORMATION PROCESSING SYSTEM 360 FORMATTED FILE SYSTEM (N--ETC(U)  
JUN 75

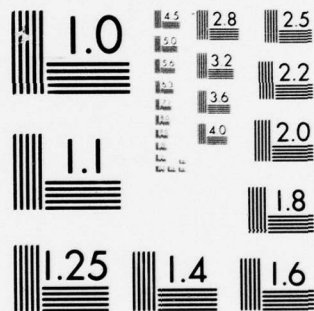
**NMCSSC-CSM-15-74-V8-1/2/3**

NL

2 OF 2.

AD  
A056212

END  
DATE  
FILMED  
8 -78  
DDC



MICROCOPY RESOLUTION TEST CHART  
NATIONAL BUREAU OF STANDARDS-1963-A

## JOB PREPARATION

```
//DCAT      EXEC  PGM=IDCAMS
//DDMAST DD    DSN=AMASTCAT,DISP=SHR
//DDNIP     DD    UNIT=3330,VOL=SER=xxxxxx,DISP=OLD
//SYSPRINT DD  SYSOUT=A
//SYSIN     DD   *
      DEFINE USERCATALOG -
          (NAME (NIESM) VOLUME (xxxxxx) -
           FILE (DDNIP) TRACKS (133,8)) -
          DATA (RECORDS (23,8))
```

/\*

In the preceding example the DDMAST statement points to the master catalog, needed to receive a catalog entry for the NIPS user catalog; and the DDNIP statement points to the disk pack which will contain the NIPS user catalog. The NIPS user catalog was named NIPSM. A complete discussion concerning the creation of user catalogs is contained in the OS/VS Access Methods Services manual (GC26-3836).

### 6.1.2 Defining a Cluster for a VSAM File

Prior to running a NIPS job which creates a new VSAM file, the user must first define the cluster that represents that VSAM file via IDCAMS. In defining the cluster, the user specifies the name of the file, the volume that contains the file, the key size and location, etc. The following JCL could be used to define a cluster for the VSAM file VSAM.SECOND:

```
//DCLUS     EXEC  PGM=IDCAMS
//STEP1CAT DD  DSN=NIESM,DISP=SHR
//DDNIP     DD    UNIT=3330,VOL=SER=xxxxxx,DISP=OLD
//SYSPRINT DD  SYSOUT=A
//SYSIN     DD   *
      DEFINE CLUSTER -
          ( NAME (VSAM.SECOND) VOL (xxxxxx) -
           KEYS (21,1)  BUFFERSPACE (13312) -
           FREESPACE (20 20) CYLINDERS (2)) -
          DATA (RECORDSIZE (100,1000) -
           CONTROLINTERVALSIZE (4096)) -
          INDEX ( CONTROLINTERVALSIZE (1024) -
           REPLICATE IMEL) -
          CAT (NIPSM)
```

/\*



## JOB PREPARATION

In the preceding example the STEPCAT statement points to the NIPS user catalog which will contain the catalog entry for the VSAM file. The DDNIP statement points to the disk pack which will contain the VSAM file.

For NIPS VSAM files the KEYS parameter is specified as follows:

KEYS (KS,1)

where KS is the keysize. The key is a minimum of 15 bytes and consists of the following:

- a. Two bytes for system control
- b. The user defined major control group
- c. The largest user or system defined set control group, a minimum of four bytes.

A complete discussion concerning the definition of a cluster is contained in the OS/VS Access Method Services Manual.

### 6.1.3 Converting an ISAM File to VSAM

The user can use the IDCAMS service routine to convert an existing ISAM file to a new VSAM file. The following JCL could be used to convert the ISAM file, ISAM.SECOND, to the VSAM file, VSAM.SECOND:

```
//REP      EXEC PGM=IDCAMS
//STEPCAT DD DSN=NIPSM,DISP=SHR
//DDVSAM DD DSN=VSAM.SECOND,DISP=OLD
//DDISAM DD DSN=ISAM.SECOND,UNIT=3330,
//          VCL=SER=xxxxxx,DISP=SHR,
//          ECB=DSORG=IS
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
           REPRC INFILE(DDISAM) -
           OUTFILE (DDVSAM)
/*
```

In the preceding example the STEPCAT statement specifies the NIPS user catalog containing the catalog entry for the

## JOB PREPARATION

VSAM file. The DDVSAM statement specifies the VSAM file and the DDISAM statement specifies the input ISAM file.

### 6.1.4 Deleting a Cluster

If it becomes necessary to delete a VSAM file, the user must use IDCAMS to delete the file. IDCAMS will scratch the file from the user's disk pack and will delete the file catalog entry in the NIPS user catalog. The following JCL could be used to delete the VSAM file, VSAM.SECOND:

```
//DCAT      EXEC PGM=IDCAMS
//STEPCAT DD  ISN=NIPSM,DISP=SHR
//DDNIP DD   UNIT=3330,VOL=SER=xxxxxx,DISP=CLD
//SYSPRINT DD SYSCUT=A
//SYSIN DD   *
            DELETE VSAM.SECOND CATALOG (NIPSM) -
            FILE (DDNIP) PURGE
/*
```

In the preceding example the STEPCAT statement specifies the user NIPS catalog containing the catalog entry for the VSAM file to be deleted and the DDNIP statement points to the disk pack containing the VSAM file. A complete discussion of the delete procedure within IDCAMS is contained in the OS/VS Access Service Methods Manual.

### 6.2 File Maintenance (FM)

When File Maintenance (FM) is used to process a VSAM file, as either a data base, transaction source or AFR file, the user must specify the NIPS user catalog that contains the catalog entry for the VSAM file. This is done by including the VSCAT parameter on the EXEC statement. The following is an example of a completed VSCAT parameter:

```
//FMS EXEC XFM,VSCAT=NIPSM
```

#### 6.2.1 VSAM Generate

When running a VSAM generate operation, the user must specify the name of the VSAM file containing the PFI via the VSDSN parameter, and the name of the newly generated VSAM file via the NEWVSM parameter. These two file names (DSNAMES) cannot be the same, as they refer to two different

## JOB PREPARATION

VSAM files. The file specified by the NEWVSM parameter will be the newly generated file and must have been previously defined via the VSAM service routine IDCAMS (Section 6.1.2). The user must not include any of the parameters associated with an ISAM generate. The following job could be used to generate a VSAM file:

```
//GEN      EXEC XFM,VSCAT=NIPSM,LIB=PRIME,  
//          VSDSN='VSAM.PFT.PRIME',  
//          NEWVSM='VSAM.PRIME'  
//FM.SYSIN DD *  
$FMS/GEN,PRIME,XXX,,DISK,CARD  
.  
.    GENERATE TRANSACTIONS  
.  
/*
```

In the above example, the term DISK on the \$FMS control card denotes that the data base will be either ISAM or VSAM.

### 6.2.2 VSAM Update

When running a VSAM update, the user specifies the name of the VSAM file with the VSDSN parameter. The user must not include any of the symbolic parameters associated with an ISAM update. The following job could be used to update a VSAM FILE:

```
//UPD      EXEC XFM,VSCAT=NIPSM,  
//          VSDSN='VSAM.PRIME',  
//          LIB=PRIME  
//FM.SYSIN DD *  
$FMS/UPD,PRIME,XXX,,DISK,CARD  
.  
.    UPDATE TRANSACTIONS  
.  
/*
```

In the above example, the term DISK on the \$FMS control card denotes that the data base is either ISAM or VSAM.

### 6.2.3 VSAM Transactions

When an FM generate or update is to be performed using a VSAM file as the transaction source, the user must code

## JOB PREPARATION

the FM.TRANS DD override statement which specifies the DSNNAME, DISP and AMP parameters. The AMP parameter must be coded as follows:

```
AMP='AMORG'
```

Additional subparameters may be specified as set forth in the OS/VS VSAM Programmer's Guide (GC26-3838).

If more than one VSAM file is to be used as the transaction source, each file must be identified by a separate DD statement. The DD statements must be named ISTRANxx, where xx is a unique identifier for each DD statement. The same parameters specified for a single VSAM file on the FM.TRANS DD statement must be specified for each VSAM file on the corresponding ISTRANxx DD statement. The \$FMS/xxx control card would specify the transaction source as ISAM. The following job could be used to update a VSAM file using a VSAM file as input transactions.

```
//UPD      EXEC XFM,VSCAT=NIPSM,
//          VSDSN='VSAM.PRIME',
//          LIB=PRIME
//FM.TRANS DD DSN=VSAM.SECOND,DISP=SHR,
//          AMP='AMORG'
//FM.SYSIN DD *
$FMS/UPD,PRIME,ZZZ,,DISK,ISAM
/*
```

### 6.2.4 Processing SAM Files With a VSAM FFT/LS

It is possible for the user to elect to perform a SAM file generate using the FFT and logic statements from a VSAM file. This can be accomplished by specifying the VSCAT, VSDSN, and SAMCUT parameters along with the other parameters needed to perform a SAM run. The following job could be used to generate a SAM file using a VSAM FFT:

```
//SGEN      EXEC XFM,VSCAT=NIPSM,
//          VSDSN='VSAM.PRIME',
//          SAMCUT=,VSMCUT='SER=xxxxxx'
//FM.SYSIN DD *
$FMS/GEN,PRIME,YYY,,TAPE,CARD
.
.      GENEFATE TRANSACTIONS
```



## JOB PREPARATION

/\*

The user can also perform a SAM update while using the FFT and logic statements from an existing VSAM file. This is accomplished by preparing the EXEC statement for a normal SAM update while including the VSCAT parameter and overriding the FM.ISAMWCRK DD statement to point to the VSAM file. The user must provide the DSNAME, DISP, and AMP parameters on the ISAMWCRK DD statement. The following job could be used to update a SAM file using the FFT and logic statements from a VSAM file:

```
//SUPD      EXEC XFF,VSCAT=NIPSM,
//          SAM=PRIME,LIB=PRIME,
//          VSAM='SER=xxxxxx',
//          VSMOUT='SER=xxxxxx',
//          SAMOUT=
//FM.ISAMWORK DD DSN=VSAM.PRIME,DISP=SHR,
//          AMP='AMORG'
//FM.SYSIN DD *
$FMS/UPD,PRIME,XYZ,,TAPE,CARD
.
.   UPDATE TRANSACTIONS
.
/*
```

### 6.2.5 Other FM VSAM Considerations

If the user employs the AFR operator in POCL to address a VSAM file, the DD statement for the referenced VSAM file must specify the DSNAME, DISP and AMP parameters. Also the user must include the VSCAT parameter on the EXEC statement. The following DD statement could be used for an AFR referenced VSAM file:

```
//AFRVSM DD DSN=VSAM.PRIME,DISP=SHR,
//          AMP='AMORG'
```

The BSZNEW parameter used to specify output block size does not apply when generating a VSAM file, as its block size was established when the file cluster was defined via IDCAMS.



## JOB PREPARATION

When the user is performing a SAM generate with a VSAM PFT, `BSZNEW` must be used to specify the block size of the new SAM file if a block size other than 1004 is desired.

### 6.3 File Revision (FR)

File Revision (FR) will revise the format of a NIPS VSAM file as well as a SAM or ISAM file. The new PFT can be either an ISAM PFT or a VSAM PFT. When running FR against a VSAM file or using a new VSAM PFT, the user must include the `VSCAT` symbolic parameter on the `EXEC` statement to specify the NIPS user catalog.

When the user is revising an old VSAM file, the `EXEC` statement must include the `VSOLDF` parameter (specifying the old file name) and the `TRANSTYP` parameter (equal to VSAM). Additionally, the following JCL override DD statement must be included:

```
//FR.VSAM DD AMP='AMORG'
```

`TRANSTYP` parameter (equal to VSAM).

When the new PFT is a VSAM file, the user must include the `VSDSN` parameter (specifying the PFT name) on the `EXEC` statement.

The following JCL could be used to revise an old VSAM file using a new VSAM PFT:

```
//REV      EXEC XFF,VSCAT=NIPSM,  
//          VSDSN='NEW.PRIME',  
//          VSOLDF='VSAM.PRIME',  
//          TRANSTYP=VSAM,  
//          VSMOUT='SEP=xxxxxx'  
//FR.VSAM DD AMP='AMORG'  
//FR.SYSIN DD *  
           FR CONTROL CARDS  
/*
```

The block size of the new SAM file will be 1004 when the file being revised is a VSAM file, unless a different block size is specified in the `BSZNEW` parameter on the `EXEC` statement.

## JCB PREPARATION

### 6.4 File Structure (FS)

File Structure (FS) is used to structure the FFT for a new file. This new file can be either an ISAM file or a VSAM file. When the new file is VSAM, the user must have previously defined the file using the VSAM service routine IDCAMS (Section 6.1.2). To structure a VSAM file FFT, the user must include the VSCAT parameter (specifying the name of the NIPS user catalog), the VSDSN parameter (specifying the name of the VSAM file), and the NONVSM parameter (must equal 'DUMMY,'). The following JCL could be used to structure a VSAM FFT:

```
//FSSTR EXEC XFS,VSCAT=NIPSM,  
//      VSDSN='VSAM.PPT.PRIME',  
//      NONVSM='DUMMY,',  
//      LIB=PRIME  
//FS.SYSIN DD *  
        SOURCE FFT DECK  
/*
```

The BSZNEW parameter, which is normally used to specify a block size other than 1004, has no application when structuring a VSAM FFT. block size is specified when the VSAM file was defined with the IDCAMS utility.

If the structure of a VSAM FFT fails, the file will not be automatically deleted. The user must delete the VSAM file using IDCAMS (Section 6.1.4) and then redefine the file with IDCAMS prior to rerunning the FS job.

### 6.5 VSAM to SAM Procedure (XISTOS)

The procedure which is used to unload an ISAM file to SAM (XISTOS) can also be used to unload a VSAM file. To unload a VSAM file, the user must include the VSCAT parameter (specifying the NIPS catalog) and the VSDSN parameter (specifying the VSAM file name) on the EXEC statement. ISAM parameters are not permitted. The blocksize of the output SAM file will be 1004 unless overridden by use of the BSZNEW parameter on the EXEC statement. The following JCL could be used to unload a VSAM file:

## JOB PREPARATION

```
//VSTS      EXEC XISTCS,VSCAT=NIPSM,  
//          VSDSN='VSAM.PRIME',  
//          SAM=PRIME,VSAM='SER=xxxxxx'  
/*
```

### 6.6 OP

The user of a VSAM file can use the OP component of NIPS to produce reports as if the file was an ISAM file. To process a VSAM file, the user specifies the name of the VSAM file using the ISAM parameter and the NIPS user catalog with the VSCAT parameter. The user must also include the following JCL override statement:

```
//OP.DATASET DD AMP='AMORG'
```

The following JCL could be used to run a source direct OP against the VSAM.PRIME file:

```
//SD          EXEC XCESD,VSCAT=NIPSM,  
//          ISAM='VSAM.PRIME',LIB=PRIME  
//OP.DATASET DD AMP='AMORG'  
//OP.SYSIN DD *  
              CF CONTROL CARDS  
/*
```

To perform merged file processing against VSAM files, the user must specify the name of the NIPS user catalog with the VSCAT parameter and the names of the VSAM files using the ISAM, ISAM1, and ISAM2 parameters. For each VSAM file used, the user must include a JCL override DD statement to specify AMP='AMORG'. The DD statements are DATASET for ISAM, DATAFILE1 for ISAM1 and DATAFILE2 for ISAM2. Additional secondary files may be specified by including a DATAFILEx DD statement for each additional file, when x is a one digit number from 3 to 9. Each DD statement must contain the file DSNAME, DISP=SHR and AMP='AMORG'. The following JCL could be used to run a merged file source direct OP against VSAM files:

```
//SD3          EXEC XOPSD,VSCAT=NIPSM,  
//          ISAM='VSAM.PRIME',ISAM1='VSAM.SECOND',  
//          ISAM2='VSAM.THIRD',LIB=PRIME  
//OP.DATASET DD AMP='AMORG'  
//OP.DATASET DD AMP='AMORG'
```

## JOB PREPARATION

```
//OP.DATAFIL2 DD AMP='AMORG'  
//OP.DATAFIL3 DD AMP='AMORG',DSN=VSAM.FORTH,  
//          DISP=SHR  
//OP.SYSIN DD *  
          CE CONTROL CARDS  
/*
```

The only operand which must be coded for the AMP parameter is "AMORG". Additional operands can be supplied and their descriptions are contained in the OS/VS VSAM Programmer's Guide (GC26-3838).

### 6.7 QUIP

Information can be retrieved and output from VSAM files using QUIP, but only in the batch mode. To process a VSAM file using QUIP the user must specify the NIPS user catalog with the VSCAT parameter and the name of the file in the ISAM parameter. The user must also override the DATAFILE DD statement to specify AMP='AMORG'. The following JCL could be used to run a source direct QUIP against a VSAM file:

```
//QPVSMM EXEC XQUIPSD,VSCAT=NIPSM,  
// ISAM='VSAM.PRIME',LIB=PRIME  
//QUIP.DATAFILE DD AMP='AMORG'  
//QUIP.SYSIN DD *  
          QUIP SOURCE STATEMENT  
/*
```

When using IFO, the first two secondary files are specified by using the ISAM1 and ISAM2 parameters on the EXEC statement. The user must also override the DATAFIL1 and DATAFIL2 DD statements to include the AMP='AMORG' parameter. If more than two secondary files are required, each additional file must be specified by including a DATAFILx DD statement, where x is a unique one digit number. The DD statement must include the DSNNAME, DISP=SHR and AMP='AMORG'. A total of nine secondary files may be specified. The following JCL could be used to run a source direct QUIP with IFO against VSAM files.

```
//QPVSMM EXEC XQUIPSD,VSCAT=NIPSM,  
//          ISAM='VSAM.PRIME',ISAM1='VSAM.SECOND',  
//          ISAM2='VSAM.THIRD',LIB=PRIME  
//QUIP.DATAFILE DD AMP='AMORG'
```



## JOB PREPARATION

```
//QUIP.DATAFIL1 DD AMP='AMORG'  
//QUIP.DATAFIL2 DD AMP='AMORG'  
//QUIP.DATAFIL3 DD DSN=VSAM.FORTH,DISP=SHR,AMP='AMORG'  
//QUIP.SYSIN DD *  
      QUIP SOURCE STATEMENTS  
/*
```

The only operand which must be coded for the AMP parameter is 'AMORG'. Additional operands can be supplied as described in the OS/VS VSAM Programmer's Guide (QC26-3838).

## 6.8 RASP

RASP may be used to retrieve data from a VSAM file for output by CP or QUIP. To retrieve data from a VSAM file the user must specify the NIPS user catalog with the VSCAT parameter and the file name with the ISAM parameter. The user must also override the DATAFILE DD statement to include AMP='AMORG'.

When doing a merged file retrieval against VSAM files, the user must specify the names of the additional files via the ISAM1 and ISAM2 parameters and must provide overrides for the DATAFIL1 and DATAFIL2 DD statements to include AMP='AMORG'. When more than three files are being processed, the user must add a DATAFILx DD statement for each file, where x is a unique one digit number. Each DD statement must have a DSNAME, DISP=SHR and AMP='AMORG'. The following JCL could be used to run a RASP against VSAM files:

```
//PSP EXEC XRASP,VSCAT=NIPS*,  
// ISAM='VSAM.PRIME',  
// ISAM1='VSAM.SECOND',  
// LIB=PRIME  
//RASP.DATAFILE DD AMF='AMORG'  
//RASP.DATAFIL1 DD AMP='AMORG'  
//RASP.SYSIN DD *  
      RASP SOURCE STATEMENTS  
/*
```

The only operand which must be coded for the AMP parameter is 'AMORG'. Additional operands can be supplied



## JCB PREPARATION

as described in the OS/VS VSAM Programmer's Guide (GC26-3838) .

### 6.9 SAM to VSAM Procedure (XSTOIS)

The procedure which is used to load an ISAM file from a SAM file (XSTOIS) can also be used to load a VSAM FILE. To load a VSAM file, the user must first define the file using the VSAM service routine IDCAMS (Section 6.1.2) . To use the XSTOIS procedure for VSAM, the user must specify the NIPS user catalog with the VSCAT parameter, specify the file name with the VSDSN parameter and void the ISAM files by specifying NONVSM='DUMMY,'. The following JCL could be used to load a VSAM file:

```
//STOVS EXEC XSTOIS,VSCAT=NIPSM,  
//      VSDSN='VSAM.PRIME',  
//      NONVSM='DUMMY,',  
//      SAM=PRIME,VSAM='ser=xxxxxx'  
/*
```

### 6.10 XDUMPLIE

To run an XDUMPLIE job against a VSAM file, the user must specify the NIPS user catalog with the VSCAT parameter and the file name with the ISAM parameter. The user must also override the DATAFILE DD statement to include AMP='AMORG'. The following JCL could be used to run an XDUMPLIE job against a VSAM file:

```
//DMP EXEC XDUMPLIE,VSCAT=NIPSM,  
//      ISAM='VSAM.PRIME'  
//UTDMP.DATAFILE DD AMP='AMORG'  
//UTDMP.SYSIN DD *  
//      XDUMPLIE CONTROL CARDS  
/*
```

Additional AMP operands may be specified as described in the OS/VS VSAM Programmer's Guide (GC26-3838) .

### 6.11 XCLASS

To change the classification of a VSAM file, the XCLASS procedure would be used. The user must specify the NIPS user catalog with the VSCAT parameter and the file name with

## JOB PREPARATION

the ISAM parameter. The DATAFILE DD statement must be overridden to include AMP='AMORG'. The following JCL could be used to change the classification of a VSAM file:

```
//CLS      EXEC XCLASS,VSCAT=NIPSM,ISAM='VSAM.PRIME'  
//CLASS.DATAFILE DD AMP='AMORG'  
//CLASS.SYSIN DD *  
          CLASSIFICATION CARD  
/*
```

Additional AMP operands may be specified as described in the OS/VS VSAM Programmer's Guide (GC26-3838).

### 6.12 XUTFSCAN

To use a VSAM file as the input for a field scan job, the user specifies the NIPS user catalog with the VSCAT parameter and the file name with the ISAM parameter. The DATAFILE DD statement must be overridden to include AMP='AMORG'. The following JCL could be used to run an XUTFSCAN involving a VSAM file:

```
//      EXEC XUTFSCAN,VSCAT=NIPSM,ISAM='VSAM.PRIME',  
//      LIE=PRIME,TRANS=PRIMET  
//UTFLOSCN.DATAFILE DD AMP='AMORG'  
//UTFLOSCN.SYSIN DD *  
          SCAN CARDS  
/*
```

Additional AMP operands may be specified as described in the OS/VS VSAM Programmer's Guide (GC26-3838).

### 6.13 XSP

The XSP procedure can be used to perform index specification on a VSAM file. The user must specify the NIPS user catalog with the VSCAT parameter and the file name with the ISAM procedure. The NEWFILE DD statement must be overridden to include AMP='AMORG'. The following JCL could be used to index a VSAM file.

```
//XSPV      EXEC XSE,VSCAT=NIPSM,  
//      ISAM='VSAM.PRIME',XINDEX=PRIME,  
//      XVOI='SER=xxxxxx',LIB=PRIME  
//XSP.NEWFILE DD AMP='AMORG'
```

## JOB PREPARATION

```
//XSP.SYSIN DD *  
    INDEX CARDS  
/*
```

Additional AMP operands may be specified as described in the OS/VS VSAM Programmer's Guide (GC26-3838).

### 6.14 XTRDISK

When the user is unloading an Index Data Set for a VSAM file and the option STAT=YES is specified, the NIPS user catalog must be specified with the VSCAT parameter and the file name with the ISAM parameter. The DATAFILE DD statement must be overridden to include AMP='AMORG'. The following JCL could be used to unload the index of a VSAM file:

```
//STPS EXEC XTRDISK,VSCAT=NIPSM,STAT=YES,  
// ISAM='VSAM.PRIME',XFNAME=PRIME  
//XTR.DATAFILE DD AMP='AMORG'  
/*
```

Additional AMP operands may be specified as described in the OS/VS VSAM Programmer's Guide (GC26-3838).

### 6.15 XKA

The XKA procedure can be used to perform keyword analysis on a VSAM file. The user must specify the NIPS user catalog with the VSCAT parameter and the file name with the ISAM parameter. The DATAFILE DD statement must be overridden to include AMP='AMORG'. The following JCL could be used to perform keyword analysis on a VSAM file:

```
//KEY EXEC XKA,VSCAT=NIPSM,  
// ISAM='VSAM.PRIME',LIB=PRIME  
//XKA.DATAFILE DD AMP='AMORG'  
//XKA.SYSIN DD *  
    XKA CONTROL CARDS  
/*
```

Additional AMP operands may be specified as described in the OS/VS VSAM Programmer's Guide (GC26-3838).

<u>PARAMETER</u>	<u>DEFINITION</u>
CHKSP	This parameter defines the space allocated for the data set (CHECKDD DD card) utilized in checkpoint/restart for the 'execute only' procedures.
CYLOFL	This specifies amount of cylinder overflow for a data base generated by FM or loaded to disk from tape.
DEN	This specifies the tape density.
DNSMCUT	This defines the name of the SAM file generated by this run. If this parameter is not used, the data set name in the FMS control card is used.
EROPT	This specifies the error option to be selected when an I/O error occurs while reading SAM transactions in FM.
GEN	This is used to control the allocation of the NEWFILE data set in FM.
Please note that all references to secondary indexing functions include keyword indexing functions. Keyword stop word tables and dictionaries are stored as members in either private user libraries or in the common DUMMY.FILEL library. To obtain keyword functions, use secondary indexing procedures and specify symbolic parameters to include the libraries that contain keyword data (see File Libraries, Section 2.3.3).	
INDEX	This defines the number of cylinders allocated to the index of a new ISAM data file.
#ISAM ISAM1 ISAM2	These define ISAM or VSAM data file names.
JOBLIE	This defines the partitioned data set containing program load modules.
JOBMAC	This defines the partitioned data set containing generative code macros.
LAB	This defines label types for sequential files.



<u>PARAMETER</u>	<u>DEFINITION</u>
CHKSP	This parameter defines the space allocated for the data set (CHECKDD DD card) utilized in checkpoint/restart for the 'execute only' procedures.
CYLOFL	This specifies amount of cylinder overflow for a data base generated by FM or loaded to disk from tape.
DEN	This specifies the tape density.
DNSMOUT	This defines the name of the SAM file generated by this run. If this parameter is not used, the data set name in the FMS control card is used.
EROPT	This specifies the error option to be selected when an I/O error occurs while reading SAM transactions in FM.
GEN	This is used to control the allocation of the NEWFILE data set in FM.
Please note that all references to secondary indexing functions include keyword indexing functions. Keyword stop word tables and dictionaries are stored as members in either private user libraries or in the common DUMMY.FILELIB library. To obtain keyword functions, use secondary indexing procedures and specify symbolic parameters to include the libraries that contain keyword data (see File Libraries, Section 2.3.3).	
INDEX	This defines the number of cylinders allocated to the index of a new ISAM data file.
#ISAM ISAM1 ISAM2	These define ISAM or VSAM data file names.
JOBLIE	This defines the partitioned data set containing program load modules.
JOBMAC	This defines the partitioned data set containing generative code macros.
LAB	This defines label types for sequential files.



<u>PARAMETER</u>	<u>DEFINITION</u>
Labin	This defines the label type for the SAM data file for the XTRIISK procedure when requesting the statistics option.
LIB LIB1 LIB2	These define user-library file names.
LIBDISP	This defines the disposition of user library(s).
NAME	This specifies the name of the member to be listed from a source library.
NBRBLK	This defines the number of blocks required to hold the Index Data Set.
MCTSP	This defines the space allocation for temporary storage of new data records and data records whose key or length has changed during an FM update.
MODLIB	This defines the name of the user's subroutine library for the subroutine loader.
NDISP	This defines the disposition of the ISAM data file in XPS and XSTOIS procedures, and the disposition of the new file in 360 to 1410 conversion.
NEWFFT	This defines the name of the new FFT for FR.
#NEWVSM	This defines the name of the newly generated VSAM file in FM.
NRMDSF	This parameter defines the normal disposition for the work data sets and the CHECKDD data set utilized in checkpoint/restart in case of an ABEND.
#NONVSM	This is used to control allocation of ISAM data files when processing VSAM files in FS and ISTOS.

PARAMETER

DEFINITION

CDISP

This defines the disposition of the old data file in the X360CON and the XISTCS procedures.

OLDLAE

This defines the label parameter of the 1410 data file during data conversion.

<u>PARAMETER</u>	<u>DEFINITION</u>
OLDSAM	This defines the name of the old SAM file in the XISTOS procedure.
OLDVSAM	This defines the volume for the old SAM file in the XISTOS procedure.
OSDISP	This defines the disposition of the old SAM file in the XISTOS procedure.
OVFLOW	This defines the number of cylinders allocated to the overflow area for a new ISAM data file.
PRIME	This defines the number of cylinders allocated to the prime area for a new ISAM data file.
PTFJOBL	This defines the data set name of a partitioned data set containing the PTF load modules.
PTFJOEM	This defines the data set name of the partitioned data set containing PTF generative code macros.
QDF	This defines the name of the RASP qualified data file if other than a temporary name is required.
QDFSP	This defines the space allocation for the RASP qualified data file.
QDISP	This defines the disposition of the RASP qualified data file and qualified record table.
QRT	This defines the name of the RASP qualified record table.
QRTSP	This defines the space allocation for the RASP qualified record table.
SAM SAM1 SAM2	These define SAM data file names.
SEQNO	This parameter defines the data set's position with respect to other data sets on the volume.

<u>PARAMETER</u>	<u>DEFINITION</u>
SDISP	This defines the disposition of the user's source library.
SLAB	This defines the label type for sort work tapes.
SAMOUT	This is used to control the allocation of the FMSAMOUT and FMNDATA data sets in FM.
SORTSP	This defines the cylinder allocation for disk sort work areas.
STAT	This requests the statistics option for XTEDISK index utility.
STG	This defines the unit type for sort work areas.
SOURCL	This defines the user's source library names.
TELK	This defines the blocksize for the File Analysis Statistics transaction data set.
TDISP	This defines the disposition for the File Analysis Statistics transaction data set.
TRANS	This defines the File Analysis Statistics transaction data file name.
TRANSP	This defines the space allocation for temporary storage of update transactions with the sort key and logic statement name appended.
#TRANTYP	This defines to the FM component the data set containing records to be used as transactions during execution of FR. Default value is ISAM. The value for this parameter is either ISAM, SAM or VSAM and denotes the access method associated with the file being revised.

PARAMETER

DEFINITION

TRBUENO      This defines the number of buffers used by  
              some of the transaction temporary data sets.

TRCH         This specifies the TRTCH DCB parameter for  
              seven-track tapes.



<u>PARAMETER</u>	<u>DEFINITION</u>
UCHK	This parameter defines the unit type for the CHECKDD DD statement in the 'execute only' procedures.
UFPT	This defines the unit type of the new FFT for FR.
UISAM	This defines the unit type for all ISAM files.
ULIB ULIB1 ULIB2	These define the unit type for all user libraries.
UQDP	This defines the unit type for the RASP qualified data file.
UQRT	This defines the unit type for the RASP-qualified record table.
USAM	This defines the unit type for all SAM files.
USMMCT	This defines the unit type of the tape on which the new file will be written if major control fields are changed during an FM SAM or FR run.
USMOUT	This defines the unit type of the tape on which the new file will be written if no major control fields are changed during an FM SAM or FR run.
USOUPCL	This defines the unit type for the user source library.
U1410	This defines the unit for a 1410 data base.
VCHK	This parameter defines the volume for the CHECKDD DD statement in the 'execute only' procedures.
VFPT	This defines the volume of the new FFT for FR.

<u>PARAMETER</u>	<u>DEFINITION</u>
VISAM VISAM1 VISAM2	These define the volume for all ISAM files.
VLIB VLIB1 VLIB2	These define the volume for user libraries.
VINDEX VOVFLOW VPRIME	These define the volumes on which the respective portions (INDEX, OVFLOW, PRIME) of the new ISAM data set will be allocated during an FM generate run.
VSOURCE	This defines the volume of the user's source library.
VQDP	This defines the volume for the RASP-qualified data file.
VQRT	This defines the volume for the RASP-qualified record table.
VSAM VSAM1 VSAM2	These define the volume for all SAM files.
VSCAT	This defines the NIPS user catalog for VSAM processing.
#VSDSN	This defines the VSAM file being processed in FM, PS, ISTOS, and STCIS and the new VSAM FFT in PR.
VSMCT	This defines the volume on which the new file will be written if major control fields are changed during an FM SAM or PR run.
VSMOUT	This defines the volume on which the new file will be written if no major control fields are changed during an FM SAM or PR run.
#VSOLF	This defines the VSAM file being revised in PR.
V1410	This defines the volume for a 1410 data base.

<u>PARAMETER</u>	<u>DEFINITION</u>
XDISP	This defines the disposition of the Index Data Set.
XFDISP	This parameter describes the status of the "FROM" Index Data Set, and indicates what is to be done with it after termination of UTNDXTPR.
XPNAME	This parameter defines the name given to the "FROM" Index Data Set for UTNDXTPR.
XPUNIT	This UTNDXTPR parameter is used to specify information about the input unit used by the "FROM" Index Data Set.
XPVOL	This UTNDXTPR parameter provides information about the volume on which the "FROM" Index Data Set resides.
XINDEX	This defines the name of the Index Data Set.
XINDEX1 XINDEX2	This defines additional Index Data Sets to be used in a merge file retrieval.
XTDISP	This parameter describes the status of the "TO" Index Data Set, and indicates what is to be done with it after termination of CTNDXTPR.
XTNAME	This parameter defines the name given to the "TO" Index Data Set for UTNDXTPR.
XTVOL	This UTNDXTPR parameter provides information about the volume on which the "TO" Index Data Set resides.
XUNIT	This is the unit type for the Index Data Set.
XVOL XVOL1 XVOL2	These define the volume for Index Data Sets.

Appendix E  
INPUT SOURCE DNAME

<u>COMPONENT</u>	<u>STEPNAME</u> . <u>DDNAME</u>
RASP	RASP.SYSIN
CP	CP.SYSIN
PM	PM.SYSIN
PS	PS.SYSIN
PR	PR.SYSIN
QUIP	QUIP.SYSIN
TABGEN	TAB.SYSIN
SUBLDR	SUB.SYSIN
SAM TO ISAM	STP1 (No Input Source)
ISAM TO SAM	STP1 (No Input Source)
360CON	GC.SYSIN
1410CON	GC.SYSIN
UTQRTQDF	QRTQDF.SYSIN
UTDMPLIB	UTDMP.SYSIN
UTFLDSCN	UTF.SYSIN
UTSUBCHK	SUBCHK.SYSIN
UTCLASS	CLASS.SYSIN
UTSOURC	SCURC.SYSIN
XSP	UTXSP.SYSIN
UTNDXKAN	XKA.SYSIN
XKM	XKM.SYSIN
XKA	XKA.SYSIN

## Appendix C

### PROCEDURES DESCRIPTIONS

<u>PROCEDURE</u>	<u>DESCRIPTION</u>
XOP	This procedure is normally only used to publish the results of a RASP retrieval run or to structure an RIT.
XOPSD	This procedure is used to publish directly from a data base or to structure a RIT.
XPM	This procedure is used to perform all file maintenance functions.
XPS	This procedure is used to structure a data base.
XRASP	This procedure is used to retrieve and sort data from one or more data files.
XISTOS	This procedure is used to convert an ISAM or VSAM data file to a SAM tape data file or copy a SAM tape data file to a SAM tape data file.
#XSTOIS	This procedure is used to convert a SAM tape data base to an ISAM or VSAM disk data base.
XSUBLDR	This procedure is used to link-edit a user subroutine (which has already been assembled) into a file library with the necessary control information to load the subroutine at execution time.
XPMEX	This procedure is used to perform file maintenance updates using stored logic statements.
XRASPEX	This procedure is used to retrieve and sort data from one or more data files using stored retrievals.
XOPEX	This procedure is used to publish the results of a RASP retrieval run using a stored RIT.
XOPSDX	This procedure is used to publish directly from a data file using a stored RIT.



<u>PROCEDURE</u>	<u>DESCRIPTION</u>
XTABGEN	This procedure is used to generate tables and place them in a user library.
X1410CON	This procedure is used to convert a NIPS 360 FFS data base to a 1410 NIPS data base.
X360CON	This procedure is used to convert a 1410 NIPS data base to a NIPS 360 FFS data base.
XFR	This procedure is used to revise a NIPS 360 FFS data base to a new format.
XQRTQDF	This procedure is used to create a NIPS 360 FFS data file from the answer file produced by RASP.
XQUIP	This procedure is used to publish the results from a RASP retrieval run.
XQUIPSD	This procedure is used to retrieve, sort, and output directly from a NIPS 360 FFS data file.
XDMP LIB	This procedure is used to print logic statements and/or report names from a NIPS 360 FFS data file.
XSUBCHK	This procedure is used to test user-written subroutines.
XCLASS	This procedure is used to change the classification on any NIPS 360 FFS data file.
XUTFSCAN	This procedure is used to scan component source statements and count the number of data field references in each source statement.
XUTSOURC	This procedure is used to add, replace, delete, or list source members from a library.
XSP	This procedure is used to either generate or update an Index Data Set based on an ISAM data file.

## JOB PREPARATION

See the IBM System/360 Operating System JobControl Language Reference, Form GC28-6704 for a description of the use of return codes.

# Appendix E

## NIPS PROCEDURE DD STATEMENT USAGE

### PROCEDURE: XFS (File Structuring)

#STPCAT	Defines the NIPS user catalog for VSAM processing.
STEPLIB	Defines the NIPS system library.
SLIB	Defines the user and system libraries.
#NEWVSM	Defines the VSAM FFT output by PS phase 2. The file must have been previously defined by IDCAMS.
NEWFILE	Defines the FFT output of PS phase 2. It is an indexed sequential data set. If the FFT is to be saved, the ISAM, VISAM, and DISP symbolic parameters must be coded accordingly.
SORTLIB	Defines the S/360 Operating System's sort library.
SORTWK01 through SORTWK06	Defines work data set used by SORT. They must all be the same unit type.
SORTIN	Defines the output of PS phase 1 and input to SORT. Contains the temporary FFT entries, one per FIELD or GROUP card.
SORTOUT	Defines the output of SORT and input to PS phase 2. Contains the temporary FFT entries sorted into alphabetical order on FIELD and GROUP names.
SYSOUT	Defines the error list from SORT. Currently set up as a dummy data set.
SYSUDUMP	ABEND dump printer output.
SYSPRINT	Defines the printed output of FS. Contains FFT listings and any error messages.

## JOB PREPARATION

DELNEW	Defines the PPT output that is scratched if the PS run was unsuccessful due to errors.
SOURCEPT	Defines a printer output data set used for source listings.
SOURCELIB	Defines a library used to store source material.
SYSUT1	Temporary data set for INDEX Specification statements to be processed by the INDEX Specification processor, IXSP, which builds descriptor (D) records for the file.
SYSIN	Defines the input source statements for PS. This statement must be supplied by the user as //FS.SYSIN DD *.

### PROCEDURE: XFR (File Revision)

The following DD statements appear in the FR and FFTST steps of this procedure. For a description of all other DD statements, refer to the XPM procedure.

#STPCAT	Defines the NIPS user catalog for VSAM processing.
STEPLIB	Defines the NIPS system library.
SYSOUT	Defines a printer output data set for sort messages.
SYSPRINT	Defines a printer output data set.
SYSUDUMP	ABEND dump printer output.
SYSPUNCH	Defines a punch output data set.
#OLDVSM	Defines the VSAM data file to be revised.
#VSMFILE	Defines the data set containing the new VSAM FFT.
OLDFILE	Defines the ISAM data file to be revised.

## JOB PREPARATION

SAMFILE	Defines the SAM data file to be revised.
DATAFILE	Defines a data set containing the new ISAM FFT.
TEDRCDS	Defines a temporary data set for generated TDD cards.
POOLFCDS	Defines a temporary data set for generated FCCL statements.
PRVSNA	Defines a temporary data set containing first 24 generated logic statements.



## JOB PREPARATION

FRVSNE Defines a temporary data set containing second group of 24 generated logic statements if needed.

FRVSN0 Defines the temporary data set containing a third group of 24 generated logic statements if needed.

FRGENCD Defines the temporary data set FMS control card for file generation steps; passed to GENT and GEN0 steps.

SORTLIB Defines the S/360 Operating System's Sort library.

SORTIN Defines the temporary data set containing intermediate work records to be sorted.

SORTWK01 through SORTWK06 Defines the work data sets used by SORT. They must all be the same unit type.

SORTOUT Defines the temporary data set for sorted intermediate work records.

SYSIN Defines the input source statements for FR. This statement must be supplied by the user as //FR.SYSIN DD \*.

TRANSTYP Defines the DD card to be used for input transaction to FM.

ISAM Defines to FM where update transactions may be found. Entries in this DD correspond to the CIDEFILE DD.

SAM Defines to FM where update transactions may be found. Entries in this DD correspond to the SAMFILE DD.

#VSAM Defines to FM where update transactions may be found. Entries in this DD correspond to the OLIVSM DD.

## JOB PREPARATION

### PROCEDURE: XFM, XFMEX (File Maintenance)

*STEPCAT	Defines the NIPS user catalog for VSAM processing.
STEPLIB	Defines the NIPS system library.
SYSOUT	Printer output for sort messages.
SYSPRINT	Printer output for remainder of FM.
SYSLIST	Printer output for assembler listing of logic statements.
SYSUDUMP	ABEND dump printer output.
*VSMFILE	Defines the VSAM data file to be processed by FM.
*NEWVSM	Defines the VSAM data file generated by an FMS/GEN. This data file must have been previously defined by IDCAMS.
DATAFILE	Defines the indexed sequential data file to be processed by FM.
NEWFILE	Defines the indexed sequential data file created during an FM generation run. The user normally codes the INDEX, PRIME, and CVPLCW symbolic parameters accordingly on the EXEC card.
FMCOMM	Defines a temporary data set used for section communication data.
FMFLUD	Defines a temporary data set used for logic statement compilation and contains formatted images of the user's input statements.
FMTRANS	Defines a temporary data set containing the images of card transactions, if any.
FMLABELS	Defines a temporary data set containing lists of ECOL instruction labels used in the logic statements.

## JOB PREPARATION

#PMSTAT	Printer output for statistics gathered during the run.
PMCM	Defines a temporary data set containing an intermediate form of the logic statement control records.
PMCMFILE	Defines a temporary data set containing the logic statement control member records in their final format.
FMLITFIL	Defines a temporary data set containing the literals produced for logic statement compilation.
SYSLIB	Identifies the library containing those macros necessary to generate logic statements (Gen Code macros).
PMLEIN	Defines the temporary data set containing the output of the compiler.
SYSMOD	Defines the data set containing the logic statement CSECT load modules.
TEMLSREC	Defines the temporary data set containing the list of the names of temporary logic statements.
SORTLIB	Defines the S/360 Operating System's sort library.
SORTIN TAPEIN	Defines the data sets containing the unsorted update records.
SORTWK01 through SORTWK06	Defines the work data sets residing on disk and used by sort.
SORTOUT TAPEOUT	Defines the data sets containing the sorted update records.
TAPEWK01 through TAPEWK04	Defines the work data sets residing on tape and used by sort.
FMLEFILE	Defines the temporary data set containing the linkage editor control statements.

## JOB PREPARATION

**#RECSIN** Defines a temporary data set containing new data records and subsets to be used as input to a data record sort. These records will eventually be merged into the data file to form the new SAM data file.

**RECSOUT** Defines a temporary data set containing the sorted new data records and subsets.

**RECSWK01 through RECSWK06** Defines the disk sort work areas for the sort of the data records.

**TRANS** Defines the transaction input file. This DD statement must be overridden by the user to identify a single disk or transaction data set. For multiple transaction sources see section 3.1.

**FMSETTBL** Defines the temporary data set containing a list of the periodic set numbers to be accessed by the FM run.

**#FMAUXOP** Defines a temporary data set containing all the auxiliary output records. All auxiliary records, regardless of device they are to be written to, are first written to FMAUXOP.

**PUNCH** Defines the punch data set for the user's punched output.

**AUX1 through AUX5** Defines the data sets for tape or disk auxiliary output. The user must override these DD statements to describe the data sets he wishes to produce.

**FMTAUX** Defines a temporary data set containing output for a second printer output.

**SLIB** Defines a temporary partitioned data set containing temporary logic statements, and the user and system libraries.

**OMMACS** Defines a temporary data set containing the compiler input of those macros generated for Ordinary Maintenance (OM).

## JOB PREPARATION

**PMMACRO** Defines the temporary data set containing the input to the PM compiler.

**PMOUTMAC** Defines the temporary data set containing the Gen Code macro statements that will be input to the PM compiler.



## JOB PREPARATION

GENFILE	Defines a temporary data set used as a work area for the OM compiler.
LITFILE	Same properties as PMITFIL but dedicated for OM use only.
EDITFILE	Defines a data set used as a work area by the OM compiler.
FMNPLUD	Same properties as FMFLUD but dedicated for OM use only.
FMTEFILE	Defines a temporary data set containing extended TDD formats for OM.
FMAEFIL	Defines a temporary data set containing extended ADD statement records for OM.
FMSODANO	This DD statement prevents the update of a data set simultaneously by SODA (Source Data) and FM.
NFLEBP	Defines a temporary data set containing error messages reflecting those errors detected by the NFI editing steps.
NFLMACS	Defines a temporary data set containing NFL generated macro prototypes with their corresponding parameters; it is used as input to the macro generator.
ASSEMIN	Defines two concatenated data sets containing generated macros and those values specified in OM DEFINE statements; all used as input to the assembler.
ISAMWORK	Defines a temporary ISAM data set containing the PFI and logic statement library during sequential file processing.
SAMFILE	Defines the sequential data set to be processed by FM.
FMNDATA	Defines the sequential output data file when record key changes occurred during sequential file processing.

## JOB PREPARATION

FMSAMOUT	Defines the output sequential data file that is produced by sequential file processing without key changes.
STAT	Defines the temporary data set used by the statistics capability.
#TRANST	Defines a temporary data set containing the File Analysis Statistics Transactions data file.
SOURCPRT	Defines a printer output data set used for source listings.
SOURCLIB	Defines a library used to store source material.
SYSIN	Defines the input source statements for FM. This statement must be supplied by the user as //FM.SYSIN DD *.
IXTRANS	This defines the temporary SAM data set used to hold the Index Transaction Records for input to Index Maintenance.
XINDEX	This defines the Index Data Set.

## JOB PREPARATION

### Procedure: XRASP, XRASPEX (Retrieval and Sort Processor)

#STPCAT	Defines the NIPS user catalog for VSAM processing.
STEPLIB	Defines the NIPS system library.
DATAFILE	Defines the indexed sequential data files
DATAFIL1	to be interrogated by RASP.
DATAFIL2	
SAMFILE	Defines the sequential data files to be interrogated
SAMFILE1	by RASP.
SAMFILE2	
QDFILE	Defines the sequential data set generated by RASP containing the retrieval source statements and the subset of qualified records from the retrieval.
SORTOUT	Used by S/360 sort. At the end of retrieval proper, this data set is the Qualifying Record Table (QRT) generated during retrieval time.
SOFTIN	Defines the input data set (QRT) to S/360 sort.
SORTWK01 through SORTWK06	Defines the work data sets used by S/360 sort.
SORTLIB	Defines the S/360 Operating System's sort library.
SYSLMCD	Defines a temporary data set which first contains RASP source statements and finally compiled retrievals. If permanent retrievals are specified, they are copied from SYSIMCD to the data set defined by the SLIB DD statement.
SLIB	Defines a temporary partitioned data set containing temporary retrievals, the user library and system libraries.
DLIB	Defines the user's primary file library.

## JOB PREPARATION

Procedure: XCP, XOPSD, XCFEX, XOPSDLEX (Output Processor)

#STEPCAT	Defines the NIPS user catalog for VSAM processing.
STEPLIB	Defines the NIPS system library.
DATAFILE DATAFIL1 DATAFIL2	Defines the indexed sequential data files processed by OP.
SAMFILE SAMFILE1 SAMFILE2	Defines the sequential data files processed by OP.
DLIB TLIB	Permanent RITs will be stored on one of these data sets.
SYSPRINT SYSOUT	Defines printer output data sets.
SYSUDUMP	AEEND dump printer output.
SYSUT1 SYSUT2 SYSUT3	Defines temporary work data sets.
SYSPUNCH	Defines a temporary data set containing the object modules resulting from the assembly.
SYSLIB	Defines a data set containing the Gen Code macros used to generate a RIT.
SYSLMOD	Defines an output library for temporary data sets.
SYSTPRT	Defines the printer output data set for the assembler and linkage editor listings.
OPIWCF	Defines a temporary data set containing CF supervisor control cards. Created by OPBEGIN and used by OPCTLPRC.
OPSTST	Defines a temporary data set containing FIT specification decks. Created by OPBEGIN and used by OPTAG.

## JOB PREPARATION

OPCREATE	Defines a temporary data set containing the RIT create cards and pointers to the corresponding specification deck. Created by OPBEGIN and used by OPTAG.
OPCOMREC	Defines a temporary data set containing communications for OPTAG. Created by OPBEGIN.
STRUCT	Same as CESTST.
INSTS	Defines a temporary data set containing the macro calls and instructions comprising a RIT. Created by OPTAG and used by the assembler.
LCF	Defines a temporary data set containing the linkedit control cards for permanent RIT.
LCT	Same as LCF but for temporary RITs.
LITSTP	Defines a temporary data set containing macro definitions generated by OPTAG defining the communications CSECTs of the RITs that have been structured.
NAMDEF	Defines a temporary data set containing a name-definition string defining the attributes, in coded form, of field/group names.
LITAB	Defines a temporary data set containing a table of literals defined in a RIT. Created and used by OPTAG.
RGTA	Defines a temporary data set containing a table in internal format of the functions of a RIT. Created and used by OPTAG.
CGC	Defines a temporary data set containing a table, by RIT, of CSECT ID used for constructing the link-edit control files.
ERRTAE	Defines a temporary data set containing a table of error codes recognized in the editing of the RIT specification decks.



## JOB PREPARATION

**SYSIN** Defines the input source deck and must be supplied by the user as //SUB.SYSIN DD \*.

### Procedure: XSTOIS (File Load Utility)

**#STEP CAT** Defines the NIPS user catalog for VSAM processing.

**STEPLIB** Defines the NIPS system library.

**#VSMFILE** Defines the VSAM data file to be loaded. This data file must have been previously defined by IDCAMS.

**DATAFILE** Defines the indexed sequential data set to be created.

**SAMFILE** Defines the sequential data set used to create the ISAM data set.

**SYSPRINT** Defines a printer output data set.

**SYSUDUMP** ABEND dump printer output.

### Procedure: XISTOS (File Unload Utility)

**#STEP CAT** Defines the NIPS user catalog for VSAM processing.

**STEPLIB** Defines the NIPS system library.

**#VSMFILE** Defines the VSAM data set to be unloaded.

**DATAFILE** Defines the indexed sequential data set to be unloaded.

**SAMFILE** Defines the sequential data set to be copied.

**SYSPRINT** Defines a printer output data set.

**SYSUDUMP** ABEND dump printer output.

**SAMOUT** Defines the sequential data set to be created.

## JOB PREPARATION

### Procedure: X1410CON (360 to 1410 Data Conversion)

STEPLIB	Defines the NIPS system library.
SYSPRINT SYSOUT	Defines printer output data sets.
SYSUDUMP	ABEND dump printer output.
DATAFILE	Defines the NIPS 360 FFS ISAM data set to be converted to a 1410 data file.
NEWFILE	Defines the sequential 1410 data file generated by this utility.
SYSIN	Defines the 1410 FFI object deck and must be supplied by the user as //GO.SYSIN DD *.

### Procedure: X360CON (1410 to 360 Data Conversion)

SYSPRINT SYSOUT	Defines printer output data sets.
SYSUDUMP	ABEND dump printer output.
DATAFILE	Defines the NIPS 360 FFS ISAM FFI.
FILE1410	Defines the 1410 data file.
NEWFILE	Defines the NIPS 360 FFS sequential data set generated by this utility.
SYSIN	Defines the 1410 FFI object deck and must be supplied by the user as //GO.SYSIN DD *.

## JOB PREPARATION

### Procedure: XQRTQDF

STEPLIB	Defines the NIPS system library.
SYSPRINT	Defines a printer output data set.
SYSUDUMP	ABEND dump printer output.
QRTFILE	Defines a data set containing sorted pointers into the QDFILE created by RASP.
QDFILE	Defines a data set containing those data records qualified by RASP.
SAMFILE	Defines a sequential data set which has all the properties of a S/360 SAM data set.
SYSIN	Defines the QRTQDF source input statements. This statement must be supplied by the user as //QRTQDF.SYSIN DD *.

## JOB PREPARATION

Procedure: XQUIP, XQUIPSD (Quick Inquiry Processor)

*STPCAT	Defines the NIPS user catalog for VSAM processing.
STEPLIB	Defines the NIPS system library.
SYSUT1	Defines temporary work data sets used for structuring the query, and for Index processing.
SYSUT2	
SYSUT3	
SYSUT4	
SLIB	Defines user and system libraries.
DATAFIL1	Defines the indexed sequential data sets accessed by QUIP as secondary files in Interfile Output.
DATAFIL2	
SAMFILE	Defines the sequential data set accessed by QUIP in source direct mode.
SYSPPRINT	Defines a printer output data set.
QUERYQUE	Defines the temporary data set containing temporary queries if the queries are to be stored on QUERYLIB after execution.
PB	Defines a temporary data set containing those data records qualified by QUIP in source direct mode.
KEY	Defines a temporary data set containing the sort keys generated by QUIP for those data records which qualified in source direct mode.
SORTWK01 through SORTWK04	Defines work areas used by the internal QUIP sort when in source direct mode.
SYSUDUMP	ABEND dump printer output.
SCURCPT	Defines a printer output data set used for source listings.
SOURCELIB	Defines a library used to store source material.

## JOB PREPARATION

SYSIN	Defines the QUIP source statement input. This statement must be supplied by the user when QUIP is run in the batch partition. It is coded as //QUIP.SYSIN DD *.
QDFILE	Defines the data set containing the data records qualified by RASP.
QRTFILE	Defines the RASP-generated data set containing sorted pointers (QRT) into the QDFILE.
SIAT	Defines the temporary data set used by the statistics capability.
XINDEX	This defines the Index Data Set.



## JOB PREPARATION

### Procedure: XDMPLIB

*STERCAT	Defines the NIPS user catalog for VSAM processing.
STERLIB	Defines the NIPS system library.
SYSPRINT	Defines a printer output data set.
SYSDUMP	ABEND dump printer output.
DATASET	Defines an indexed sequential data set.
ISAMWORK	Defines a temporary ISAM data set containing the FFT and logic statement library during sequential file processing.
SAMPLE	Defines the sequential data set to be processed by UTDMPLIB.
SYSIN	Defines the input source control card for UTDMPLIB. This statement must be supplied by the user as //UTDMP.SYSIN DD *.

## JOB PREPARATION

### Procedure: YSUBCHK (Subroutine Check)

STEPLIB	Defines the NIPS system library.
SYSUDUMP	ABEND dump printer output.
SYSPPINT	Defines a printer output data set.
SYSOUT	Defines a printer output data set.
SLIB	Defines user and system libraries.
SYSIN	Derives the input source deck and must be supplied by the user as //SUBCHK.SYSIN DD *.

## JOB PREPARATION

### Procedure: YCLASS (File Classification Change)

#STEPCAT	Defines the NIPS user catalog for VSAM processing.
STEPLIB	Defines the NIPS system library.
SYSPPINT	Defines a printer output data set.
SYSUDUMP	ABEND dump printer output.
DATAFILE	Defines the ISAM data file to be processed.
SAMFILE	Defines the sequential data set to be processed. This will be both the input and output data set for sequential processing on disk.
UTSAMOUT	Defines the output sequential data file that is produced when tape input is used.
SYSIN	Defines the CLASS source input card. This statement must be supplied by the user as //CLASS.SYSIN DD *.

## JOB PREPARATION

### Procedure: XUTESCAN (Field Scan)

#STEPCAT	Defines the NIPS user catalog for VSAM processing.
SYSPRINT	Defines a printer output data set.
SYSUDUMP	ABEND dump printer output.
DATAFILE	Defines the ISAM data file.
SAMFILE	Defines the SAM data file.
ISAMWORK	Defines temporary ISAM data set if a SAM data file is input.
SYSUT2	Defines partitioned data set containing members to be scanned.
TRANST	Defines the data set for output transactions.
SYSIN	Defines input stream. This card must be supplied by the user as //UTF.SYSIN DD *.
SOURCEPPT	Defines a printer output data set.
SOURCELIB	Defines a library used to store source material.
SYSIN	Defines the input to the UTSOURCE utility. This is not overridden if a LIST operation is desired. This statement is overridden if library update is to be performed.

## JOB PREPARATION

### Procedure: XSP (Index Specification)

*STECAT	Defines the NIPS user catalog for VSAM processing.
STEPLIB	Defines the NIPS system library.
NEWFILE	Defines the indexed sequential data file created during an FM run.
XINDEX	Defines the Index Data Set.
SORTWK01 through SORTWK06	Defines the S/360 Operating System's Sort work data sets.
SORTLIB	Defines the S/360 Operating System's Sort Library.
SYSOUT	Defines a printer output data set for Sort messages.
SYSPPRINT	Defines a printer output data set.
SOURCEPPR	Defines a printer output data set for Index Specification messages.
SYSUDUMP	ABEND dump printer output.
SLIB	Defines the user library containing subroutines/tables.
SAMFILE	Defines a sequential data file
UISAMOUT	Defines the updated sequential data file.
ISAMWORK	Defines the temporary ISAM data set containing the FFT during sequential file processing in Index Specification.



## JOB PREPARATION

### Procedure: XTRDISK (Index Transfer - Unload)

#STEPCAT	Defines the NIPS user catalog for VSAM processing.
STEPLIB	Defines the NIPS system library.
SYSPPRINT	Defines the printed output of XTRDISK. Contains any diagnostic messages.
SYSUDUMP	AEEND dump printer output.
INDEXSAM	Defines the output sequential Index Data Set.
INDEXDAM	Defines the input, direct access, disk-resident Index Data Set.
DATAFILE	Defines the ISAM data file corresponding to the index data set.
SAMFILE	Defines the sequential access data file corresponding to the index data set.

## JOB PREPARATION

Procedure: XTIRTAPE (Index Transfer - Load)

STEPLIB	Defines the NIPS system library.
SYSPRINT	Defines the printed output of XTIRTAPE. Contains any diagnostic messages.
SYSDUMP	ABEND dump printer output.
INDEXSAM	Defines the input, sequential version of an Index Data Set.
XINDEX	Defines the output, disk-resident, direct access Index Data Set.

## JOB PREPARATION

Procedure: XKA (Keyword Analysis)

#STEDCAT	Defines the NIPS user catalog for VSAM processing.
STEPLIB	Defines the NIPS system library.
DATAFILE	Defines the indexed sequential data file to be analyzed.
SAMFILE	Defines the sequential data files to be analyzed.
SLIB	Defines the user library containing subroutine and/or tables.
SYSPPINT	Defines a printer output data set.
SOURCLIB	Defines a library used to store source material.
SOURCEPPT	Defines a printer output data set used for source listings.
SYSUDUMP	Defines a printer output data set for ABEND dump output.
SYSOUT	Defines a printer output data set containing S/360 sort messages.
SORTLIB	Defines the S/360 Operating System sort library.
SORTWK01 thru SORTWK04	Defines the work data sets used by S/360 sort.
KANWK01	Defines a temporary work data set.
KMDPPT	Defines a printer output data set.
SYSIN	Defines the input user control statement data set for UTNEXKAN. This statement must be supplied by the user as //XKA.SYSIN DD *.

## JOB PREPARATION

**#Procedure:** XUTODE

STEPLIB	Defines the NIPS system library.
SYSUDUMP	Defines a printer output data set for ABEND.
SNAPSHOT	Defines a printer output data set for debug.
SOURCEPRT	Defines a printer output data set for source listings.
ODDPRT	Defines a printer output data set for diagnostics.
SYSPRINT	Defines a printer output data set.
SLIB	Defines the library where the user compiled format will be stored.
SOURCELIB	Defines a library used to store source material.

## JOB PREPARATION

<u>Procedure:</u>	UTNDXKMD
STEPLIB	Defines the NIPS System Library.
SLIB	Defines the user library containing tails.
KMDWK01 thru KMDWK04	Defines UTNDYKMD work data sets.
KMDPBT	Defines message and display device, UTNEXKMD.
SYSOUT	Sort message output device.
SYSPEINT	Printer output device.
SORTLIB	S/360 OS Sort Library.
SORTWK01 thru SOPTWK06	Sort work data sets.
SYSUDUMP	ABEND dump device.
SYSIN	SYSIN device.



JOB PREPARATION

Appendix F  
PROCEDURE LISTINGS

107

CH-3

PRECEDING PAGE BLANK-NOT FILMED

## NIPS 360 FFS

## PROCEDURES

```

//XDMPLIB  PROC  A=A,BSZFILE=, 00000100
//          CL=',' ,CL1=',' ,DEN=,ISAM='DUMMY.FILE', 00000200
//          JOBLIB='FFS.JOBLIB',LAB=SL,RGN=60K, 00000300
//          PTFJOBL='PTF.JOBLIB', 00000400
//          SAM='DUMMY.FILE',STG=NIPW,TRCH=, 00000500
//          UISAM='(2314,P)',USAM='(TAPE9,,DFFER)', 00000600
//          VISAM=,VSAM= 00000700
//          00000800
// * CHARLES W. HICKISCH MAJOR,USA PROJ CODE=763NIPS BRANCH 431 00000900
// * DATE=MARCH 1,1974 00001000
//          00001100
//UTDMP EXEC PGM=UTDMPLIB,REGION=&RGN 00001200
//STEPLIB DD DSN=&PTFJOBL,DISP=SHR 00001300
//          DD DSN=&JOBLIB,DISP=SHR 00001400
//SYSPRINT DD SYSOUT=(&A,&CL) 00001500
//SYSUDUMP DD SYSOUT=(&A,&CL1) 00001600
//DATAFILE DD DSNAME=&ISAM,UNIT=&UISAM,VOLUME=&VISAM,DISP=SHR 00001700
//ISAMWORK DD UNIT=&STG,DCB=DSORG=IS,SPACE=(CYL,(10)) 00001800
//SAMFILE DD DSNAME=&SAM.S,UNIT=&USAM,VOLUME=&VSAM,DISP=(SHR,KEEP), 00001900
//          LABEL=(,&LAB),DCB=(RECFM=VB,LRECL=1000,BLKSIZE=&BSZFILE, 00002000
//          TRCH=&TRCH,DEN=&DEN) 00002100

```

## NIPS 360 FFS

## PROCEDURES

```

//XFM  PROC  A=A,AUXBUFN=2,AUXSP=6,BLKSIZE=560,B=B,          00000100
//*  MODIFIED DEC 76  RTF  WHCC VSAM  A3798                    00000110
//    VSCAT='DUMMY.FILE',VSDSN='DUMMY.FILE',NEWVSM='DUMMY.FILE', 00000120
//*  A3798                                                    00000150
//    BSZFILE=,BSZNEWF=,                                       X00000200
//    CL=',' ,CL1=',' ,CL2=',' ,CYLCFL=1,DEN=,                00000300
//    EROPT=ABE,GEN='DUMMY,' ,                                00000400
//    INDEX=1,ISAM='DUMMY.FILE',                               00000500
//    INXSP=1,JOBLIB='FFS.JOBLIB',JOBMAC='FFS.JOBMACRO',      00000600
//    LAB=SL,LIB='DUMMY.FILE',LIB1='DUMMY.FILE',              X00000700
//    MCTSP=5,NRRBLK=200,QVFLCW=1,PRIME=5,PGA=98K,            00000800
//    PTFJOBL='PTF.JOBLIB',PTFJOBM='PTF.JOBMACRO',            00000900
//    SAM='DUMMY.FILE',SAMOUT='DUMMY,' ,SDISP=SHR,SCRTSP=10,   00001000
//    SOURCL='DUMMY.FILE',STG=NIPW,TDISP=MCD,TRANS='&&TRANS', 00001100
//    TRANSP=200,TRBUFN=4,TRCH=,UISAM='(2314,P)',             00001200
//    ULIB=2314,ULIB1=2314,USAM='(TAPE9,DEFER)',              00001300
//    USOURCL=2314,UTRANS=NIPW,VINDEX='REF=*.DATAFILE',        00001400
//    VISAM=,VLIB=,VLIB1=,VOVFLCW='REF=*.DATAFILE',           00001500
//    VPRIME='REF=*.DATAFILE',VSAM=,VSMCT=,VSMOUT=,VSOURCL=,  00001600
//    VTRANS=,XDISP=SHR,XINDEX='DUMMY.FILE',                  00001700
//    XUNIT=2314,XVOL=                                         00001800
//*                                                            00001900
//*  CHARLES W. HICKISCH MAJOR,USA  PROJ CODE=763NIPS  BRANCH=431 00002000
//*  DATE=MARCH 1, 1974                                       00002100
//*                                                            00002200
//FM  EXEC  PGM=FM,REGION=GRGN                                00002300
//STEP CAT DD DSN=&VSCAT,DISP=SHR                              00002310
//STEP LIB DD DSN=&PTFJOBL,DISP=SHR                             00002400
//    DD DSN=&JOBLIB,DISP=SHR                                   00002500
//FMSTAT DD SYSOUT=(&A,&CL)                                     00002600
//SYSOUT DD SYSOUT=(&A,&CL)                                     00002700
//SYSLIST DD SYSOUT=(&A,&CL)                                     00002800
//SOURCPRT DD SYSOUT=(&A,&CL)                                    00002900
//SYSUDUMP DD SYSOUT=(&A,&CL1)                                   00003000
//SYSPRINT DD SYSOUT=(&A,&CL2)                                   00003100
//VSMFILE DD DSN=&VSDSN,DISP=SHR                                00003110
//NEWVSM DD DSN=&NEWVSM,DISP=SHR                                00003120
//DATAFILE DD DSN=&ISAM,UNIT=&UISAM,VOLUME=&VISAM,DISP=SHR     00003200
//NEWFILE DD &GEN.DCB=(DSORG=IS,CYLCFL=&CYLCFL,BLKSIZE=&BSZNEWF), X00003300
//    DISP=(,KEEP,DELETE),SPACE=(CYL,&INDEX),                  X00003400
//    DSN=&ISAM.A(INDEX),UNIT=&UISAM,VOLUME=&VINDEX              00003500
//    DD &GEN.DCB=*.NEWFILE,SPACE=(CYL,&PRIME),                X00003600
//    DSN=&ISAM.A(PRIME),UNIT=&UISAM,VOLUME=&VPRIME,            X00003700
//    DISP=(,KEEP,DELETE)                                       00003800
//    DD &GEN.DCB=*.NEWFILE,SPACE=(CYL,&CVFLCW),                X00003900
//    DSN=&ISAM.A(CVFLOW),UNIT=&UISAM,VOLUME=&VOVFLOW,          X00004000
//    DISP=(,KEEP,DELETE)                                       00004100
//FMCOMM DD UNIT=&STG,SPACE=(TRK,(1,1))                          00004200
//FMFLUD DD UNIT=&STG,SPACE=(CYL,(&MCTSP,5))                     00004300
//FMTRANS DD UNIT=&STG,SPACE=(CYL,(&AUXSP,5)),DCB=BUFN=&TRBUFN 00004400
//FMLABELS DD UNIT=&STG,SPACE=(CYL,(1,1))                        00004500
//FMCM DD UNIT=&STG,SPACE=(TRK,(20,5))                           00004600
//FMCMFILE DD UNIT=&STG,SPACE=(TRK,(20,5))                       00004700
//FMFLITFIL DD UNIT=&STG,SPACE=(TRK,(20,5))                     00004800
//SYSLIB DD DSN=&PTFJOBM,DISP=SHR                                00004900
//    DD DSN=&JOBMAC,DISP=SHR                                   00005000
//    DD DSN=&SYS1.MACLIB,DISP=SHR                               00005100

```

```

//FMLEIN DD UNIT=&STG,SPACE=(TRK,(25,5)) 00005200
//SYSLMOD DD UNIT=&STG,SPACE=(CYL,(4,1,10)),DCB=SYSL.LINKLIB, X00005300
// LABEL=EXPDT=66366 00005400
//TEMLREC DD UNIT=&STG,SPACE=(TRK,(20,5)) 00005500
//SORTLIB DD DSN=SYSL.SORTLIB,DISP=SHR 00005600
//SORTIN DD UNIT=&STG,SPACE=(TRK,(&TRANSP,150)),DCB=BUFNO=&TRBUFN 00005700
//SORTWK01 DD SPACE=(CYL,(&SORTSP),,CONTIG),UNIT=(&STG,SEP=SORTIN) 00005800
//SORTWK02 DD SPACE=(CYL,(&SORTSP),,CONTIG), X00005900
// UNIT=(&STG,SEP=SORTWK01),SEP=SCRTWK01 00006000
//SORTWK03 DD SPACE=(CYL,(&SORTSP),,CONTIG), X00006100
// UNIT=(&STG,SEP=SCRTWK02),SEP=SCRTWK02 00006200
//SORTWK04 DD SPACE=(CYL,(&SORTSP),,CONTIG), X00006300
// UNIT=(&STG,SEP=(SORTWK01,SORTWK03)), X00006400
// SEP=(SORTWK01,SORTWK03) 00006500
//SORTWK05 DD SPACE=(CYL,(&SORTSP),,CONTIG), X00006600
// UNIT=(&STG,SEP=(SORTWK02,SORTWK04)), X00006700
// SEP=(SORTWK02,SORTWK04) 00006800
//SORTWK06 DD SPACE=(CYL,(&SORTSP),,CONTIG), X00006900
// UNIT=(&STG,SEP=(SORTWK03,SORTWK05)), X00007000
// SEP=(SORTWK03,SORTWK05) 00007100
//SORTOUT DD DISP=(OLD,PASS),DSNAME=*.SCRTIN,VOLUME=REF=*.SCRTIN, *00007200
// DCB=(LRECL=1500,BLKSIZE=1504,RECFM=VB,BUFNO=2) 00007300
//FMLEFILE DD DSN=*.SCRTWK04,VOLUME=REF=*.SCRTWK04,DISP=(OLD,PASS) 00007400
//FMMACRO DD DSN=*.SCRTWK05,VOLUME=REF=*.SCRTWK05,DISP=(OLD,PASS) 00007500
//FMOUTMAC DD DSN=*.SCRTWK06,VOLUME=REF=*.SCRTWK06,DISP=(OLD,PASS) 00007600
//TAPEIN DD DSN=*.SCRTIN,VOLUME=REF=*.SCRTIN,DISP=(OLD,PASS) 00007700
//TAPEOUT DD DSN=*.SCRTIN,VOLUME=REF=*.SCRTIN,DISP=(OLD,PASS), *00007800
// DCB=(LRECL=1500,BLKSIZE=1504,RECFM=VB) 00007900
//RECSIN DD DSN=*.FMFLUD,VOLUME=REF=*.FMFLUD,DISP=(OLD,PASS) 00008000
//RECSOUT DD DSN=*.RECSIN,VOLUME=REF=*.RECSIN,DISP=(OLD,PASS), X00008100
// DCB=(LRECL=1000,BLKSIZE=1004,RECFM=VB) 00008200
//RECSWK01 DD DSN=*.SCRTWK01,VOLUME=REF=*.SCRTWK01,DISP=(OLD,PASS) 00008300
//RECSWK02 DD DSN=*.SCRTWK02,VOLUME=REF=*.SCRTWK02,DISP=(OLD,PASS) 00008400
//RECSWK03 DD DSN=*.SCRTWK03,VOLUME=REF=*.SCRTWK03,DISP=(OLD,PASS) 00008500
//RECSWK04 DD DSN=*.SCRTWK04,VOLUME=REF=*.SCRTWK04,DISP=(OLD,PASS) 00008600
//RECSWK05 DD DSN=*.SCRTWK05,VOLUME=REF=*.SCRTWK05,DISP=(OLD,PASS) 00008700
//RECSWK06 DD DSN=*.SCRTWK06,VOLUME=REF=*.SCRTWK06,DISP=(OLD,PASS) 00008800
//TRANS DD DUMMY,DISP=OLD,DCB=(EROPT=&EROPT,BUFNO=&TRBUFN) 00008900
//FMSETTBL DD UNIT=&STG,SPACE=(TRK,(1,1)) 00009000
//FMAUXOP DD DSN=*.FMTRANS,VOLUME=REF=*.FMTRANS,DISP=(OLD,PASS), 00009100
// DCB=(LRECL=1000,BLKSIZE=1004,BUFNO=&AUXBUFN) 00009200
//PUNCH DD SYSOUT=&B,DCB=BUFNO=&AUXBUFN 00009300
//AUX1 DD DUMMY,DCB=(RECFM=VB,BLKSIZE=1004,LRECL=1000,BUFNO=&AUXBUFN) 00009400
//AUX2 DD DUMMY,DCB=(RECFM=VB,BLKSIZE=1004,LRECL=1000,BUFNO=&AUXBUFN) 00009500
//AUX3 DD DUMMY,DCB=(RECFM=VB,BLKSIZE=1004,LRECL=1000,BUFNO=&AUXBUFN) 00009600
//AUX4 DD DUMMY,DCB=(RECFM=VB,BLKSIZE=1004,LRECL=1000,BUFNO=&AUXBUFN) 00009700
//AUX5 DD DUMMY,DCB=(RECFM=VB,BLKSIZE=1004,LRECL=1000,BUFNO=&AUXBUFN) 00009800
//FMTAUX DD UNIT=&STG,SPACE=(TRK,(10,5)),DCB=BUFNO=&AUXBUFN 00009900
//SLIB DD DSN=*.SYSLMOD,DISP=(OLD,PASS),VOLUME=REF=*.SYSLMOD 00010000
// DD DSN=&LIB.L,UNIT=&ULIB,VOLUME=&VLIB,DISP=SHR 00010100
// DD DSN=&LIB1.L,UNIT=&ULIB1,VOLUME=&VLIB1,DISP=SHR 00010200
// DD DSN=&JOBLIB,DISP=SHR 00010300
//SOURCLIB DD DISP=&SDISP,DSN=&SOURCL.L,VOL=&VSOURCL,UNIT=&USOURCL 00010400
//OMMACS DD UNIT=&STG,SPACE=(CYL,(5,1)) 00010500
//GENFILE DD UNIT=&STG,SPACE=(CYL,(3,1)) 00010600
//LITFILE DD UNIT=&STG,SPACE=(CYL,(3,1)) 00010700
//EDITFILE DD UNIT=&STG,SPACE=(CYL,(3,1)) 00010800

```



## NIPS 360 FFS

## PROCEDURES

```

//FMNFLUD DD UNIT=&STG,SPACE=(CYL,(5,1)) 00010900
//FMTEFILE DD UNIT=&STG,SPACE=(TRK,(5,1)) 00011000
//FMAEFILE DD UNIT=&STG,SPACE=(TRK,(5,1)) 00011100
//FMSODANQ DD DISP=SHR,DSNAME=&ISAM.C,VOLUME=REF=&*.STEPLIB 00011200
//NFLERR DD DSNAME=&*.SORTWK04,VOLUME=REF=&*.SORTWK04,DISP=(MOD,PASS) 00011300
//NFLMACS DD DSNAME=&*.SORTWK03,VOLUME=REF=&*.SORTWK03,DISP=(OLD,PASS) 00011400
//ASSEMIN DD DSNAME=&*.FMLITFIL,VOLUME=REF=&*.FMLITFIL,DISP=(OLD,PASS) 00011500
// DD DSNAME=&*.SORTWK06,VOLUME=REF=&*.SORTWK06,DISP=(OLD,PASS) 00011600
//ISAMWORK DD UNIT=&STG,DCB=DSCRG=IS,SPACE=(CYL,(10)) 00011700
//SAMPFILE DD DSNAME=&SAM.S,UNIT=&USAM,VOLUME=&VSAM,DISP=(SHR,KEEP), X00011800
// LABEL=(,&LAB),DCB=(RECFM=VB,LRECL=1000,BLKSIZE=&BSZFILE,X00011900
// TRTCH=&TRCH,DEN=&DEN) 00012000
//FMSAMOUT DD &SAMOUT.DCB=(RECFM=VB,LRECL=1000,BLKSIZE=&BSZNEW, X00012100
// DEN=&DEN,TRTCH=&TRCH),DISP=(NEW,KEEP), 00012200
// LABEL=(,&LAB), 00012300
// UNIT=&USAM,VOLUME=(PRIVATE,RETAIN,&VSMOUT) 00012400
//FMNDATA DD &SAMOUT.DCB=&*.FMSAMOUT,DISP=(,KEEP),LABEL=(,&LAB), X00012500
// UNIT=&USAM,VOLUME=(PRIVATE,RETAIN,&VSMOUT) 00012600
//STAT DD DSN=&FMCM,VOL=REF=&*.FMCM,DISP=(OLD,PASS) 00012700
//TRANST DD DSN=&TRANS,VOL=&VTRANS,UNIT=&LTRANS,DISP=&TDISP, 00012800
// SPACE=(TRK,1) 00012900
//INDEXPRT DD SYSOUT=(&A,&CL) 00013000
//IXTRANS DD DISP=(NEW,PASS),UNIT=&STG,SPACE=(CYL,(&INXSP,1)) 00013100
//XINDEX DD DSN=&XINDEX.X,DISP=(&XDISP,KEEP),UNIT=&XUNIT,VOL=&XVOL, 00013200
// DCB=(BLKSIZE=&BLKSIZE,RECFM=F,KEYLEN=4,DSCRG=DA), X00013300
// SPACE=(&BLKSIZE,&NBRBLK) 00013400

```



## NIPS 360 FFS

## PROCEDURES

```

//XFR      PROC  A=A,AUXSP=6,B=B,BLKSIZE=560,BSZFILE=,BSZNEW=,      X00000100
//*        MODIFIED DEC 76   RTF   WHCC VSAM   A3798                00000110
//          VSCAT='DUMMY.FILE',VSDSN='DUMMY.FILE',VSCLOF='DUMMY.FILE', 00000120
//*        A3798                00000150
//          CL=' ',CL1=' ',CL2=' ',      X00000200
//          DEN=,INXSP=1,ISAM='DUMMY.FILE',      00000300
//          JOBLIB='FFS.JOBLIB',JOBMAC='FFS.JOBMACRC',      00000400
//          LAB=SL,LIB='DUMMY.FILE',LIB1='DUMMY.FILE',      X00000500
//          MCTSP=8,NEWFFT='DUMMY.FILE',NBRBLK=200,      00000600
//          NEWUSAM='(TAPE9,,DEFER)',      00000700
//          PTFJOBL='PTF.JOBLIB',PTFJOBMAC='PTF.JOBMACRC',      00000800
//          RGN=100K,SAM='DUMMY.FILE',SORTSP=10,STG=NIPW,      00000900
//          TDISP=MOD,TRANS='&&TRANS',TRANSP=200,TRANTYP=ISAM,      00001000
//          TRCH=,UFFT='(2314,P)',UISAM='(2314,P)',      00001100
//          ULIB=2314,ULIB1=2314,      00001200
//          USAM='(TAPE9,,DEFER)',UTRANS=NIPW,VFFT=,VISAM=,      00001300
//          VLIR=,VLIR1=,VSAM=,VSMCT=,VSMOUT=,VSORT='REF=*.SORTIN',      00001400
//          VTRANS=,XDISP=SHR,XINDEX='DUMMY.FILE',      00001500
//          XUNIT=2314,XVOL=      00001600
//*                00001700
//*  CHARLES W. HICKISCH MAJOR,USA  PRCH CODE=763NIPS  BRANCH=431  00001800
//*  DATE=MARCH 1, 1974      00001900
//*                00002000
//FR        EXEC  PGM=FR,REGION=&RGN      00002100
//STEP CAT DD  DSN=&VSCAT,DISP=SHR      00002110
//STEP LIB DD  DSN=&PTFJOBL,DISP=SHR      00002200
//          DD  DSN=&JOBLIB,DISP=SHR      00002300
//SYSLIST DD  SYSOUT=(&A,&CL)      00002400
//SYSOUT DD  SYSOUT=(&A,&CL)      00002500
//SYSUDUMP DD  SYSOUT=(&A,&CL1)      00002600
//SYSPRINT DD  SYSOUT=(&A,&CL2)      00002700
//SYSPUNCH DD  SYSOUT=&B      00002800
//OLDFILE DD  DSN=&VSOLD,DISP=SHR      00002810
//VSMFILE DD  DSN=&VSDSN,DISP=SHR      00002820
//OLDFILE DD  DSN=&ISAM,UNIT=&UISAM,VOLUME=&VISAM,DISP=SHR      00002900
//DATAFILE DD  DSN=&NEWFFT,UNIT=&UFFT,VOLUME=&VFFT,DISP=SHR      00003000
//SAMFILE DD  DSN=&SAM.S,UNIT=&USAM,VOLUME=&VSAM,LABEL=(,&LAB),      C00003100
//          DCB=(RECFM=VB,LRECL=1000,BLKSIZE=&BSZFILE,DEN=&DEN,      X00003200
//          TRCH=&TRCH),DISP=(SHR,PASS)      00003300
//TDDRCDS DD  UNIT=&STG,SPACE=(CYL,(3,1))      00003400
//POOLRCDS DD  UNIT=&STG,SPACE=(CYL,(3,1))      00003500
//FRVSNB DD  UNIT=&STG,SPACE=(CYL,(5,1))      00003600
//FRVSNB DD  UNIT=&STG,SPACE=(CYL,(5,1))      00003700
//FRGENCD DD  UNIT=&STG,SPACE=(TRK,1)      00003800
//SORTLIB DD  DSN=&SYS1.SORTLIB,DISP=SHR      00003900
//FMCOMM DD  UNIT=&STG,SPACE=(TRK,(1,1))      00004000
//FMFLUD DD  UNIT=&STG,SPACE=(CYL,(&MCTSP,10))      00004100
//FMTRANS DD  UNIT=&STG,SPACE=(CYL,(&AUXSP,5))      00004200
//FMLABELS DD  UNIT=&STG,SPACE=(CYL,(1,1))      00004300
//FMCM DD  UNIT=&STG,SPACE=(TRK,(20,5))      00004400
//FMCMFILE DD  UNIT=&STG,SPACE=(TRK,(20,5))      00004500
//FMLEFIL DD  UNIT=&STG,SPACE=(TRK,(20,5))      00004600
//FMLEFIL DD  UNIT=&STG,SPACE=(CYL,(5,2))      00004700
//FMMACRO DD  UNIT=&STG,SPACE=(CYL,(8,2))      00004800
//FMOUTMAC DD  UNIT=&STG,SPACE=(CYL,(8,2))      00004900
//SYSLIB DD  DSN=&PTFJOBMAC,DISP=SHR      00005000
//          DD  DSN=&JOBMAC,DISP=SHR      00005100

```

## NIPS 360 FFS

## PROCEDURES

```

// DD DSN=SYS1.MACLIB,DISP=SHR 00005200
//FMLEIN DD UNIT=&STG,SPACE=(TRK,(25,5)) 00005300
//SYSLMOD DD UNIT=&STG,SPACE=(CYL,(4,1,10)),DCB=&JCELIB, 00005400
// LABEL=EXPDT=66366 00005500
//TEMLSREC DD UNIT=&STG,SPACE=(TRK,(20,5)) 00005600
//SORTIN DD UNIT=&STG,SPACE=(TRK,(&TRANSP,20)) 00005700
//SORTWK01 DD SPACE=(CYL,(&SORTSP),,CONTIG),UNIT=&STG 00005800
//SORTWK02 DD SPACE=(CYL,(&SORTSP),,CONTIG),UNIT=(&STG,SEP=sortwk01) 00005900
//SORTWK03 DD SPACE=(CYL,(&SORTSP),,CONTIG),UNIT=(&STG,SEP=sortwk02) 00006000
//SORTWK04 DD SPACE=(CYL,(&SORTSP),,CONTIG), 00006100
// UNIT=(&STG,SEP=(sortwk01,sortwk03)) 00006200
//SORTWK05 DD SPACE=(CYL,(&SORTSP),,CONTIG), 00006300
// UNIT=(&STG,SEP=(sortwk02,sortwk04)) 00006400
//SORTWK06 DD SPACE=(CYL,(&SORTSP),,CONTIG), 00006500
// UNIT=(&STG,SEP=(sortwk03,sortwk05)) 00006600
//SORTOUT DD DSN=*.SORTIN,VOLUME=REF=*.SORTIN,DISP=(OLD,PASS), 00006700
// DCB=(LRECL=1500,BLKSIZE=1504,RECFM=VB) 00006800
//TAPEIN DD DSN=*.SORTIN,VOLUME=&VSORT,DISP=(CLD,PASS) 00006900
//TAPEOUT DD DSN=*.SORTIN,VOLUME=&VSORT,DISP=(OLD,PASS), 00007000
// DCB=(LRECL=1500,BLKSIZE=1504,RECFM=VB) 00007100
//RECSIN DD DSN=*.FMFLUD,VOLUME=REF=*.FMFLUD,DISP=(OLD,PASS) 00007200
//RECSOUT DD DSN=*.RECSIN,VOLUME=REF=*.RECSIN,DISP=(OLD,PASS), 00007300
// DCB=(LRECL=1000,BLKSIZE=1004,RECFM=VB) 00007400
//RECSWK01 DD DSN=*.SORTWK01,VOLUME=REF=*.SORTWK01,DISP=(OLD,PASS) 00007500
//RECSWK02 DD DSN=*.SORTWK02,VOLUME=REF=*.SORTWK02,DISP=(OLD,PASS) 00007600
//RECSWK03 DD DSN=*.SORTWK03,VOLUME=REF=*.SORTWK03,DISP=(OLD,PASS) 00007700
//RECSWK04 DD DSN=*.SORTWK04,VOLUME=REF=*.SORTWK04,DISP=(OLD,PASS) 00007800
//RECSWK05 DD DSN=*.SORTWK05,VOLUME=REF=*.SORTWK05,DISP=(OLD,PASS) 00007900
//RECSWK06 DD DSN=*.SORTWK06,VOLUME=REF=*.SORTWK06,DISP=(OLD,PASS) 00008000
//TRANS DD DSN=&TRANSTYP 00008100
//VSAM DD DSN=&VSCLDF,DISP=SHR 00008110
//ISAM DD DSN=&ISAM,UNIT=&UISAM,VOLUME=&VISAM,DISP=SHR 00008200
//SAM DD DSN=&SAM.S,UNIT=&USAM,VOLUME=&VSAM,LABEL=(, &LAB), 00008300
// DCB=*.SAMFILE,DISP=(SHR,KEEP) 00008400
//ASSEMBIN DD DSN=*.FMOUTMAC,VOLUME=REF=*.FMOUTMAC,DISP=(OLD,PASS) 00008500
//FMSETTBL DD UNIT=&STG,SPACE=(TRK,(1,1)) 00008600
//FMAUXOP DD DSN=*.FMTRANS,VOLUME=REF=*.FMTRANS,DISP=(OLD,PASS), 00008700
// DCB=(LRECL=1000,BLKSIZE=1004) 00008800
//SLIB DD DSN=*.SYSLMOD,DISP=(CLD,PASS),VOLUME=REF=*.SYSLMOD 00008900
// DD DSN=&LIB.L,UNIT=&ULIB,VOLUME=&VLIB,DISP=SHR 00009000
// DD DSN=&LIB1.L,UNIT=&ULIB1,VOLUME=&VLIB1,DISP=SHR 00009100
// DD DSN=&JOBLIB,DISP=SHR 00009200
//FMSAMOUT DD UNIT=&NEWUSAM,VOLUME=(PRIVATE,RETAIN,&VSMCT), 00009300
// LABEL=(, &LAB),DISP=(NEW,KEEP), 00009400
// DCB=(RECFM=VB,LRECL=1000,BLKSIZE=&BSZNEWF,DEN=&DEN, 00009500
// TRTCH=&TRCH) 00009600
//FMNDATA DD UNIT=&NEWUSAM,VOLUME=(PRIVATE,RETAIN,&VSMCT), 00009700
// LABEL=(, &LAB),DISP=(,KEEP),DCB=*.FMSAMOUT 00009800
//STAT DD DSN=*.FMCM,VOL=REF=*.FMCM,DISP=(CLD,PASS) 00009900
//TRANST DD DSN=&TRANS,VOL=&VTRANS,UNIT=&UTRANS,DISP=&TDISP, 00010000
// SPACE=(TRK,1) 00010100
//INDEXPRT DD SYSOUT=(&A,&CL) 00010200
//IXTRANS DD DISP=(NEW,PASS),UNIT=&STG,SPACE=(CYL,(&INXSP,1)) 00010300
//XINDEX DD DSN=&XINDEX.X,DISP=(&XDISP,KEEP),UNIT=&XUNIT,VOL=&XVOL, 00010400
// DCB=(BLKSIZE=&BLKSIZE,RECFM=F,KEYLEN=4,DSORG=CA), 00010500
// SPACE=(&BLKSIZE,&NBRBLK) 00010600

```

## PROCEDURES

```

//XFS      PROC      A=*,CL1=*,*,BSZNEWF=1004,INDEX=1,
//            ISAM='DUMMY.FILE',
//            JOBLIB='FJS.JOBLIB',
//            LIB='DUMMY.FILE',LIB1='DUMMY.FILE',
//            NDISP=KEEP,CVFLCW=1,PRIME=5,RGN=60K,SDISP=SHR,
//            PTFJOBL='PTF.JOBLIB',
//            SORTSP=3,SCURCL='DUMMY.FILE',STG=NIPW,
//            UISAM=2314,ULIB=2314,ULIB1=2314,USOURCL=2314,
//            VISAM=,VLIB=,VLIB1=,VSOURCL=
//**
//** CHARLES W. HICKISCH MAJCR,USA PRJ CODE=763NIPS BRANCH=431
//** DATE=MARCH 1, 1974
//**
//**
//XFS      EXEC      PGM=FSPHASE,REGION=ERGN
//STEPLIB DD DSN=&PTFJOBL,DISP=SHR
//          DD DSN=&JOBLIB,DISP=SHR
//SLIB DD DSN=&LIB.L,DISP=SHR,VOLUME=&VLIB,UNIT=&ULIB
//          DD DSN=&LIB1.L,UNIT=&ULIB1,VOLUME=&VLIB1,DISP=SHR
//          DD DSN=&JOBLIB,DISP=SHR
//SOURCLIB DD DISP=SDISP,DSN=&SOURCL.L,VCL=&VSCURCL,UNIT=&USOURCL
//NEWFILE DD DSN=&ISAM.(INDEX),UNIT=&UISAM,VOLUME=&VISAM,
//          SPACE=(CYL,&INDEX),DCB=(DSORG=IS,BLKSIZE=&BSZNEWF),
//          DISP=(,ENDISP,DELETE)
//          DD DSN=&ISAM.(PRIME),UNIT=&UISAM,VOLUME=REF=*.NEWFILE,
//          SPACE=(CYL,&PRIME),DCB=*.NEWFILE,DISP=(,ENDISP,DELETE)
//          DD DSN=&ISAM.(OVFLOW),UNIT=&UISAM,VOLUME=REF=*.NEWFILE,
//          SPACE=(CYL,&OVFLOW),DCB=*.NEWFILE,DISP=(,ENDISP,DELETE)
//SORTLIB DD DSN=SYS1.SORTLIB,DISP=SHR
//SORTIN DD UNIT=&STG,SPACE=(CYL,(1,1)),
//          DCB=(RECFM=VB,BLKSIZE=1004,LRECL=1000)
//SORTWK01 DD SPACE=(CYL,(&SORTSP),,CONTIG),UNIT=(&STG,SEP=SORTIN)
//SORTWK02 DD SPACE=(CYL,(&SORTSP),,CONTIG),UNIT=(&STG,SEP=SORTWK01),
//          SEP=SORTWK01
//SORTWK03 DD SPACE=(CYL,(&SORTSP),,CONTIG),UNIT=(&STG,SEP=SORTWK02),
//          SEP=SORTWK02
//SORTWK04 DD SPACE=(CYL,(&SORTSP),,CONTIG),
//          UNIT=(&STG,SEP=(SORTWK01,SORTWK03)),
//          SEP=(SORTWK01,SORTWK03)
//SORTWK05 DD SPACE=(CYL,(&SORTSP),,CONTIG),
//          UNIT=(&STG,SEP=(SORTWK02,SORTWK04)),
//          SEP=(SORTWK02,SORTWK04)
//SORTWK06 DD SPACE=(CYL,(&SORTSP),,CONTIG),
//          UNIT=(&STG,SEP=(SORTWK01,SORTWK03,SORTWK05)),
//          SEP=(SORTWK01,SORTWK03,SORTWK05)
//SORTOUT DD UNIT=&STG,SPACE=(CYL,(1,1)),
//          DCB=(RECFM=VB,BLKSIZE=1004,LRECL=1000)
//SYSOUT DD DUMMY
//SYSPRINT DD SYSOUT=(&A,&CL)
//SYSUDUMP DD SYSOUT=(&A,&CL1)
//SOURCPRT DD SYSOUT=(&A,&CL)
//SYSUT1 DD DISP=(,PASS),DSN=IXSPDS,UNIT=&STG,SPACE=(TRK,(5,1))
//FSSTEP2 EXEC PGM=FSABMOD,COND=(111,NE,FS),REGION=ERGN
//STEPLIB DD DSN=&JOBLIB,DISP=SHR
//SYSOUT DD SYSOUT=(&A,&CL)
//SYSPRINT DD SYSOUT=(&A,&CL)
//DELNEW DD DSN=*.FS.NEWFILE,UNIT=&UISAM,VOLUME=REF=*.FS.NEWFILE,
//          DISP=(OLD,DELETE)

```

## NIPS 360 FFS

## PROCEDURES

```

//XISTOS PROC A=A,BSZFILE=,BSZNEW=,CC=,CL=',' ,CL1=',' ,DEN=, 00000100
// ISAM='DUMMY.FILE',JOB LIB='FFS.JOBLIB',LAB=SL,NDISP=KEEP, 00000200
// ODISP=KEEP,OLDSAM='DUMMY.FILE',CLOVSAM=, 00000300
// OSDISP=KEEP,RCN=60K,SAM='DUMMY.FILE',SECNC=1, 00000400
// PTFJOBL='PTF.JOBLIB', 00000500
// TRCH=,UISAM='(2314,P)',USAM='(TAPE9,,DEFER)', 00000600
// VISAM=,VSAM= 00000700
// * 00000800
// * CHARLES W. HICKISCH MAJOR,USA PRGJ CODE=763NIPS BRANCH=431 00000900
// * DATE=MARCH 1, 1974 00001000
// * 00001100
//STP1 EXEC PGM=UTBLDSAM,REGION=ERGN,PARM='&CC' 00001200
//STEPLIB DD DSN=&PTFJOBL,DISP=SHR 00001300
// DD DSN=&JOBLIB,DISP=SHR 00001400
//DATAFILE DD DSNAME=&ISAM,UNIT=&UISAM,VOLUME=&VISAM, 00001500
// DISP=(SHR,&ODISP,KEEP),DCB=BUFNC=5 00001600
//SAMFILE DD DSNAME=&OLDSAM.S,UNIT=&USAM,DISP=(SHR,&CSCISP), 00001700
// VOL=&CLOVSAM,DCB=(RECFM=VB,LRECL=1000,BLKSIZE=&BSZFILE, 00001800
// DEN=&DEN,TRCH=&TRCH,BUFNO=5),LABEL=(&SEQNO,&LAB) 00001900
//SAMOUT DD DSNAME=&SAM.S,UNIT=&USAM,VOLUME=&VSAM,DISP=(,ENDISP), 00002000
// DCH=(*,SAMFILE,BLKSIZE=&BSZNEW),LABEL=(&SEQNO,&LAB) 00002100
//SYSPRINT DD SYSOUT=(&A,&CL) 00002200
//SYSUDUMP DD SYSOUT=(&A,&CL1) 00002300

```



## NIPS 360 FFS

## PROCEDURES

```

//XOP  PROC  A=A,B,B,BSZFILE=,BSZFIL1=,BSZFIL2=, 00000100
//          CL=',',CL1=',',CL2=',', 00000200
//          ISAM='DUMMY.FILE',ISAM1='DUMMY.FILE',ISAM2='DUMMY.FILE', 00000300
//          JOBLIB='FFS.JOBLIB',JOBMAC='FFS.JOBMACRC', 00000400
//          LAB=SL, 00000500
//          LIB='DUMMY.FILE',LIB1='DUMMY.FILE',LIB2='DUMMY.FILE', 00000600
//          LIBDISP=SHR, 00000700
//          PTFJOBL='PTF.JOBLIB', 00000710
//          PTFJOBM='PTF.JOBMACRC', 00000720
//          QDF='&&QDFILE',QDISP=PASS,QRT='&&QRTFILE', 00000800
//          RGN=100K, 00000900
//          SAM='DUMMY.FILE',SAM1='DUMMY.FILE',SAM2='DUMMY.FILE', 00001000
//          SDISP=SHR,SOURCL='DUMMY.FILE',STG=NIPW, 00001100
//          TDISP=MCD,TRANS='&&TRANS', 00001200
//          UISAM='(2314,P)', 00001300
//          ULIB=2314,ULIB1=2314,ULIB2=2314, 00001400
//          UQDF=NIPW,UQRT=NIPW, 00001500
//          USAM='(TAPE9,,DEFER)',USOURCL=2314, 00001600
//          UTRANS=NIPW, 00001700
//          VISAM=,VISAM1=,VISAM2=, 00001800
//          VLIB=,VLIB1=,VLIB2=, 00001900
//          VQDF=,VQRT=, 00002000
//          VSAM=,VSAM1=,VSAM2=, 00002100
//          VSOURCL=,VTRANS= 00002200
//** 00002300
//** CHARLES W. HICKISCH MAJOR,USA PRCL CODE=763NIPS BRANCH=431 00002400
//** DATE=MARCH 1, 1974 00002500
//** 00002600
//OP EXEC PGM=OP,REGION=&RGN 00002700
//STEPLIB DD DSN=&PTFJOBL,DISP=SHR 00002800
//          DD DSN=&JOBLIB,DISP=SHR 00002810
//DATAFILE DD DSN=&ISAM,DISP=SHR,UNIT=&UISAM,VOLUME=&VISAM 00002900
//DATAFIL1 DD DSN=&ISAM1,DISP=SHR,UNIT=&UISAM,VOLUME=&VISAM1 00003000
//DATAFIL2 DD DSN=&ISAM2,DISP=SHR,UNIT=&UISAM,VOLUME=&VISAM2 00003100
//SAMFILE DD DSN=&SAM.S,DISP=SHR,UNIT=&USAM,VOLUME=&VSAM, X00003200
//          LABEL=(,&LAB),DCB=BLKSIZE=&BSZFILE 00003300
//SAMFILE1 DD DSN=&SAM1.S,DISP=SHR,UNIT=&USAM,VOLUME=&VSAM1, X00003400
//          LABEL=(,&LAB),DCB=BLKSIZE=&BSZFIL1 00003500
//SAMFILE2 DD DSN=&SAM2.S,DISP=SHR,UNIT=&USAM,VOLUME=&VSAM2, *00003600
//          LABEL=(,&LAB),DCB=BLKSIZE=&BSZFIL2 00003700
//DLIB DD DSN=&LIB.L,DISP=&LIBDISP,VOLUME=&VLIB,UNIT=&ULIB 00003800
//TLIB DD DSN=&LIB1.L,DISP=SHR,VOLUME=&VLIB1,UNIT=&ULIB1 00003900
//SOURCPRT DD SYSOUT=(&A,&CL) 00004000
//SOURCLIB DD DISP=&SDISP,DSN=&SOURCL.L,VCL=&VSOURCL,UNIT=&USOURCL 00004100
//SYSPRINT DD SYSOUT=(&A,&CL),DCB=(LRECL=133,BLKSIZE=665) 00004200
//SYSOUT DD SYSOUT=(&A,&CL) 00004300
//SYSUDUMP DD SYSOUT=(&A,&CL1) 00004400
//SYSUT1 DD UNIT=&STG,SPACE=(TRK,(200,20)) 00004500
//SYSUT2 DD UNIT=(&STG,SEP=SYSUT1),SPACE=(TRK,(200,20)) 00004600
//SYSUT3 DD UNIT=(&STG,SEP=(SYSUT1,SYSUT2)),SPACE=(TRK,(200,20)) 00004700
//SYSPUNCH DD UNIT=&STG,SPACE=(TRK,(50,1C)), X00004800
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=400) 00004900
//SYSLIB DD DSN=&PTFJOBM,DISP=SHR 00005000
//          DD DSN=&JOBMAC,DISP=SHR 00005010
//SYSLMOD DD UNIT=&STG,SPACE=(TRK,(50,25,5)),DCB=SYS1.LINKLIB, X00005100
//          LABEL=RETPD=0000 00005200
//SYSTPRT DD SYSOUT=(&A,&CL),DCB=(RECFM=FB,LRECL=121,BLKSIZE=605) 00005300

```



```

//OPIWCF DD UNIT=&STG,SPACE=(TRK,(2,1)),DCB=(BLKSIZE=400,LRECL=80, *00005400
// RECFM=FB) 00005500
//OPSTST DD UNIT=&STG,SPACE=(TRK,(10,2)) 00005600
//OPCREATE DD UNIT=&STG,SPACE=(TRK,(1,1)) 00005700
//OPCOMREC DD UNIT=&STG,SPACE=(TRK,(1,1)) 00005800
//INSTS DD UNIT=&STG,SPACE=(TRK,(50,5)) 00005900
//LCF DD UNIT=&STG,SPACE=(TRK,(10,5)) 00006000
//LCT DD UNIT=&STG,SPACE=(TRK,(10,5)) 00006100
//LITSTR DD UNIT=&STG,SPACE=(TRK,(50,5)) 00006200
//NAMDEF DD DSNAME=*.SYSUT1,VOLUME=REF=*.SYSUT1,DISP=(CLD,PASS) 00006300
//LITAB DD DSNAME=*.SYSUT2,VOLUME=REF=*.SYSUT2,DISP=(CLD,PASS) 00006400
//RGTAB DD DSNAME=*.SYSUT3,VOLUME=REF=*.SYSUT3,DISP=(CLD,PASS) 00006500
//CGC DD UNIT=&STG,SPACE=(TRK,(1,1)) 00006600
//ERRTAB DD UNIT=&STG,SPACE=(TRK,(8,5)) 00006700
//DCTNY DD DSNAME=*.SYSPUNCH,VOLUME=REF=*.SYSPUNCH,DISP=(OLD,PASS) 00006800
//SUBTAB DD UNIT=&STG,SPACE=(TRK,(1,1)) 00006900
//OPLGOGO DD DSNAME=*.SYSUT1,VOLUME=REF=*.SYSUT1,DISP=(CLD,PASS) 00007000
//OPLEXEX DD DSNAME=*.SYSUT2,VOLUME=REF=*.SYSUT2,DISP=(CLD,PASS) 00007100
//OPLEXDIR DD DSNAME=*.SYSUT3,VOLUME=REF=*.SYSUT3,DISP=(CLD,PASS) 00007200
//OPLINE DD SYSOUT=(&A,&CL2),DCB=LRECL=133 00007300
//OPPUNCH DD SYSOUT=GB 00007400
//SLIB DD DSNAME=*.SYSLMOD,VOLUME=REF=*.SYSLMOD,DISP=(CLD,PASS) 00007500
// DD DSNAME=*.DLIB,DISP=SHR,UNIT=&ULIB,VOLUME=&VLIB 00007600
// DD DSNAME=*.TLIB,DISP=SHR,UNIT=&ULLIB1,VOLUME=&VLIB1 00007700
// DD DSNAME=&LIB2.L,DISP=SHR,UNIT=&ULIB2,VOLUME=&VLIB2
// DD DSN=&PTFJOBL,DISP=SHR
// DD DSNAME=&JOBLIB,DISP=SHR
//SYSAIN DD DSNAME=*.LITSTR,VOLUME=REF=*.LITSTR,DISP=(CLD,PASS), 00008000
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=400,BUFNC=5) 00008100
// DD DSNAME=*.INSTS,VOLUME=REF=*.INSTS,DISP=(CLD,PASS), 00008200
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=400,BUFNC=5) 00008300
//SYSLIN DD DSNAME=*.LCF,VOLUME=REF=*.LCF,DISP=(OLD,PASS) 00008400
//SYSTIN DD DSNAME=*.LCT,VOLUME=REF=*.LCT,DISP=(OLD,PASS) 00008500
//LOADR DD DSNAME=*.SYSPUNCH,VOLUME=REF=*.SYSPUNCH,DISP=(OLD,PASS), X0008600
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=400) 00008700
//QDFILE DD DSNAME=&QDF,UNIT=&UQDF,VOLUME=&VQDF,DISP=(SHR,&QDISP,KEEP) 00008800
//QRTFILE DD DSNAME=&QRT,UNIT=&UQRT,VOLUME=&VQRT,DISP=(SHR,&QDISP,KEEP) 00008900
//STAT DD DSN=*.INSTS,VOL=REF=*.INSTS,DISP=(OLD,PASS) 00009000
//TRANST DD DSN=&TRANS,VOL=&VTRANS,UNIT=&UTRANS, 00009100
// DISP=&TDISP,SPACE=(TRK,1) 00009200

```

## NIPS 360 FFS

## PROCEDURES

```

//XOPEX  PROC  A=A,B=B,                                00000100
//              CL=' ',CL1=' ',CL2=' ',                00000200
//              CHKDSP=DELETE,CHKID='&&CHKID',CHKSP=0,CHKST=NEW, 00000300
//              ISAM='DUMMY.FILE',ISAM1='DUMMY.FILE',ISAM2='DUMMY.FILE', 00000400
//              JOBLIB='FFS.JOBLIB',                    00000500
//              LIB='DUMMY.FILE',LIB1='DUMMY.FILE',LIB2='DUMMY.FILE', 00000600
//              LIBDISP=SHR,                              00000700
//              NRMCDSP=DELETE,                          00000800
//              PTFJOBL='PTF.JOBLIB',                   00000810
//              QDF='&&QDFILE',QDISP=PASS,QRT='&&QRTFILE', 00000900
//              RGN=100K,                                 00001000
//              STG=NIPW,                                 00001100
//              TDISP=MOD,TRANS='&&TRANS',              00001200
//              UCHK=NIPW,                                00001300
//              UISAM=(2314,P)',                         00001400
//              ULIB=2314,ULIB1=2314,ULIB2=2314,        00001500
//              UQDF=NIPW,LQRT=NIPW,                    00001600
//              UTRANS=NIPW,                             00001700
//              VISAM=,VISAM1=,VISAM2=,                 00001800
//              VLIB=,VLIB1=,VLIB2=,                    00001900
//              VQDF=,VQRT=,                             00002000
//              VCHK=,VTRANS=                           00002100
//**                                                    00002200
//**  CHARLES W. HICKISCH  MAJOR,USA  PROJ CODE=763NIPS  BRANCH=431 00002300
//**  DATE=MARCH 1, 1974                                00002400
//**                                                    00002500
//OP EXEC PGM=OP,REGION=&RGN                            00002600
//STEPLIB DD DSN=&PTFJOBL,DISP=SHR                      00002700
//          DD DSN=&JOBLIB,DISP=SHR                     00002710
//DATAFILE DD DSN=&ISAM,DISP=SHR,UNIT=&UISAM,VOLUME=&VISAM 00002800
//DATAFIL1 DD DSN=&ISAM1,DISP=SHR,UNIT=&UISAM,VOLUME=&VISAM1 00002900
//DATAFIL2 DD DSN=&ISAM2,DISP=SHR,UNIT=&UISAM,VOLUME=&VISAM2 00003000
//DLIB DD DSN=&LIB.L,DISP=&LIBDISP,VOLUME=&VLIB,UNIT=&ULIB 00003100
//TLIB DD DSN=&LIB1.L,DISP=SHR,VOLUME=&VLIB1,UNIT=&ULIB1 00003200
//SOURCPT DD SYSOUT=(&A,&CL)                            00003300
//PRINTER DD SYSOUT=(&A,&CL)                             00003400
//SYSOUT DD SYSOUT=(&A,&CL)                             00003500
//SYSPRINT DD SYSOUT=(&A,&CL)                             00003600
//SYSUDUMP DD SYSOUT=(&A,&CL1)                           00003700
//OPLINE DD SYSOUT=(&A,&CL2),DCB=LRECL=133              00003800
//OPIWCF DD UNIT=&STG,SPACE=(TRK,(2,1)),DCB=(BLKSIZE=400,LRECL=80, *00003900
//          RECFM=FB),DSNAME=&CHKID.A,DISP=(, &NRMDSP,&CHKDSP) 00004000
//OPCOMREC DD UNIT=&STG,SPACE=(TRK,(1,1)),              X00004100
//          DISP=(, &NRMDSP,&CHKDSP),DSNAME=&CHKID.B        00004200
//OPLGOGO DD UNIT=&STG,SPACE=(TRK,(20,1)),              X00004300
//          DISP=(, &NRMDSP,&CHKDSP),DSNAME=&CHKID.C        00004400
//OPLEXEX DD UNIT=(&STG,SEP=OPLGOGO),SPACE=(TRK,(20,1)), X00004500
//          DISP=(, &NRMDSP,&CHKDSP),DSNAME=&CHKID.D        00004600
//OPLEXDIR DD UNIT=(&STG,SEP=(CPLGOGO,OPLEXEX)),SPACE=(TRK,(20,1)), X00004700
//          DISP=(, &NRMDSP,&CHKDSP),DSNAME=&CHKID.E        00004800
//OPPUNCH DD SYSOUT=&B                                    00004900
//SLIB DD DSN=&.DLIB,DISP=SHR,UNIT=&ULIB,VOLUME=&VLIB     00005000
// DD DSN=&.TLIB,DISP=SHR,UNIT=&ULIB1,VOLUME=&VLIB1
// DD DSN=&LIB2.L,DISP=SHR,UNIT=&ULIB2,VOLUME=&VLIB2
// DD DSN=&PTFJOBL,DISP=SHR
// DD DSN=&JOBLIB,DISP=SHR
//QDFILE DD DSN=&QDF,UNIT=&UQDF,VOLUME=&VQDF,DISP=(SHR,&QDISP,KEEP) 00005400

```

## NIPS 360 FFS

## PROCEDURES

```
//QRTFILE DD DSNAME=&QRT,UNIT=&UQRT,VOLUME=&VQRT,DISP=(SHR,&QDISP,KEEP) 00005500
//CHECKDD DD DSNAME=&CHKID,DISP=(&CHKST,&NRMDSP,&CHKDSP),UNIT=&UCHK, 00005600
//          VOLUME=&VCHK,SPACE=(CYL,(&CHKSP,1)) 00005700
//STAT DD UNIT=&STC,SPACE=(CYL,(2,1)) 00005800
//TRANST DD DSN=&TRANS,VOL=&VTRANS,UNIT=&UTRANS, 00005900
//          DISP=&TDISP,SPACE=(TRK,1) 00006000
```

## NIPS 360 FFS

## PROCEDURES

```

//XOPSD  PROC  A=A,B,BSZFILE=,BSZFIL1=,BSZFIL2=, 00000100
//          CL=',',CL1=',',CL2=',', 00000200
//          ISAM='DUMMY.FILE',ISAM1='DUMMY.FILE',ISAM2='DUMMY.FILE', 00000300
//          JOBLIB='FFS.JOBLIB',JOBMAC='FFS.JOBMACRC', 00000400
//          LAB=SL, 00000500
//          LIB='DUMMY.FILE',LIB1='DUMMY.FILE',LIB2='DUMMY.FILE', 00000600
//          LIBDISP=SHR, 00000700
//          PTFJOBL='PTF.JOBLIB',PTFJOBM='PTF.JOBMACRC', 00000800
//          RGN=100K, 00000900
//          SAM='DUMMY.FILE',SAM1='DUMMY.FILE',SAM2='DUMMY.FILE', 00001000
//          SDISP=SHR,SOURCL='DUMMY.FILE',STG=NIPW, 00001100
//          TDISP=MOD,TRANS='&&TRANS', 00001200
//          UISAM='(2314,P)', 00001300
//          ULIB=2314,ULIB1=2314,ULIB2=2314, 00001400
//          USAM='(TAPE9,,DEFER)',USOURCL=2314, 00001500
//          UTRANS=NIPW, 00001600
//          VISAM=,VISAM1=,VISAM2=, 00001700
//          VLIB=,VLIB1=,VLIB2=, 00001800
//          VSAM=,VSAM1=,VSAM2=, 00001900
//          VSOURCL=,VTRANS= 00002000
//          00002100
//          CHARLES W. HICKISCH MAJOR,USA PROJ CODE=763NIPS BRANCH=431 00002200
//          DATE=MARCH 1, 1974 00002300
//          00002400
//OP EXEC PGM=OP,REGION=&RGN 00002500
//STEPLIB DD DSN=&PTFJOBL,DISP=SHR 00002600
//          DD DSN=&JOBLIB,DISP=SHR 00002700
//DATAFILE DD DSN=&ISAM,DISP=SHR,UNIT=&UISAM,VOLUME=&VISAM 00002800
//DATAFIL1 DD DSN=&ISAM1,DISP=SHR,UNIT=&UISAM,VOLUME=&VISAM1 00002900
//DATAFIL2 DD DSN=&ISAM2,DISP=SHR,UNIT=&UISAM,VOLUME=&VISAM2 00003000
//SAMFILE DD DSN=&SAM.S,DISP=SHR,UNIT=&USAM,VOLUME=&VSAM, 00003100
//          LABEL=(, &LAB),DCB=BLKSIZE=&BSZFILE 00003200
//SAMFILE1 DD DSN=&SAM1.S,DISP=SHR,UNIT=&USAM,VOLUME=&VSAM1, 00003300
//          LABEL=(, &LAB),DCB=BLKSIZE=&BSZFIL1 00003400
//SAMFILE2 DD DSN=&SAM2.S,DISP=SHR,UNIT=&USAM,VOLUME=&VSAM2, 00003500
//          LABEL=(, &LAB),DCB=BLKSIZE=&BSZFIL2 00003600
//DLIB DD DSN=&LIB.L,DISP=&LIBDISP,VOLUME=&VLIB,UNIT=&ULIB 00003700
//TLIB DD DSN=&LIB1.L,DISP=SHR,VOLUME=&VLIB1,UNIT=&ULIB1 00003800
//SOURCPRD DD SYSOUT=(&A,&CL) 00003900
//SOURCLIB DD DISP=&SDISP,DSN=&SOURCL.L,VOL=&VSOURCL,UNIT=&USOURCL 00004000
//SYSPRINT DD SYSOUT=(&A,&CL),DCB=(LRECL=133,BLKSIZE=665) 00004100
//SYSOUT DD SYSOUT=(&A,&CL) 00004200
//SYSUDUMP DD SYSOUT=(&A,&CL1) 00004300
//SYSUT1 DD UNIT=&STG,SPACE=(TRK,(200,20)) 00004400
//SYSUT2 DD UNIT=(&STG,SEP=SYSUT1),SPACE=(TRK,(200,20)) 00004500
//SYSUT3 DD UNIT=(&STG,SEP=(SYSUT1, SYSUT2)),SPACE=(TRK,(200,20)) 00004600
//SYSPUNCH DD UNIT=&STG,SPACE=(TRK,(50,10)), 00004700
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=400,BUFNC=5) 00004800
//SYSLIB DD DSN=&PTFJOBM,DISP=SHR 00004900
//          DD DSN=&JOBMAC,DISP=SHR 00005000
//SYSLMOD DD UNIT=&STG,SPACE=(TRK,(50,25,5)),DCB=SYS1.LINKLIB, 00005100
//          LABEL=RETPD=0000 00005200
//SYSTPRD DD SYSOUT=(&A,&CL),DCB=(RECFM=FB,LRECL=121,BLKSIZE=605) 00005300
//OPIWCF DD UNIT=&STG,SPACE=(TRK,(2,1)),DCB=(BLKSIZE=400,LRECL=80, 00005400
//          RECFM=FB) 00005500
//OPSTST DD UNIT=&STG,SPACE=(TRK,(10,2)) 00005600
//OPCREATE DD UNIT=&STG,SPACE=(TRK,(1,1))

```



```

//OPCOMREC DD UNIT=ESTG,SPACE=(TRK,(1,1)) 00005700
//INSTS DD UNIT=ESTG,SPACE=(TRK,(50,5)) 00005800
//LCF DD UNIT=ESTG,SPACE=(TRK,(10,5)) 00005900
//LCT DD UNIT=ESTG,SPACE=(TRK,(10,5)) 00006000
//LITSTR DD UNIT=ESTG,SPACE=(TRK,(50,5)) 00006100
//NAMDEF DD DSN=*.SYSUT1,VOLUME=REF=*.SYSUT1,DISP=(CLD,PASS) 00006200
//LITAB DD DSN=*.SYSUT2,VOLUME=REF=*.SYSUT2,DISP=(CLD,PASS) 00006300
//RGTAB DD DSN=*.SYSUT3,VOLUME=REF=*.SYSUT3,DISP=(CLD,PASS) 00006400
//CGC DD UNIT=ESTG,SPACE=(TRK,(1,1)) 00006500
//ERRTAB DD UNIT=ESTG,SPACE=(TRK,(8,5)) 00006600
//SUBTAB DD UNIT=ESTG,SPACE=(TRK,(1,1)) 00006700
//DCTNY DD DSN=*.SYSPUNCH,VOLUME=REF=*.SYSPUNCH,DISP=(CLD,PASS) 00006800
//OPLGCGO DD DSN=*.SYSUT1,VOLUME=REF=*.SYSUT1,DISP=(CLD,PASS) 00006900
//OPLEXEX DD DSN=*.SYSUT2,VOLUME=REF=*.SYSUT2,DISP=(CLD,PASS) 00007000
//OPLEXDIR DD DSN=*.SYSUT3,VOLUME=REF=*.SYSUT3,DISP=(CLD,PASS) 00007100
//OPLINE DD SYSOUT=(EA,ECL2),DCB=LRECL=133 00007200
//OPPUNCH DD SYSOUT=EB 00007300
//SLIB DD DSN=*.SYSLMOD,VOLUME=REF=*.SYSLMOD,DISP=(CLD,PASS) 00007400
// DD DSN=*.DLIB,DISP=SHR,UNIT=ELIB,VOLUME=ELIB 00007500
// DD DSN=*.TLIB,DISP=SHR,UNIT=ELIB1,VOLUME=ELIB1 00007600
// DD DSN=ELIB2.L,DISP=SHR,UNIT=ELIB2,VOLUME=ELIB2 00007700
// DD DSN=EPFJOB.L,DISP=SHR 00007800
// DD DSN=ELIB.L,DISP=SHR 00007900
//SYSAIN DD DSN=*.LITSTR,VOLUME=REF=*.LITSTR,DISP=(CLD,PASS), 00008000
// CCB=(RECFM=FB,LRECL=80,BLKSIZE=400,BUFNO=5) 00008100
// DD DSN=*.INSTS,VOLUME=REF=*.INSTS,DISP=(CLD,PASS), 00008200
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=400,BUFNO=5) 00008300
//SYSLIN DD DSN=*.LCF,VOLUME=REF=*.LCF,DISP=(CLD,PASS) 00008400
//SYSTIN DD DSN=*.LCT,VOLUME=REF=*.LCT,DISP=(CLD,PASS) 00008500
//LOADR DD DSN=*.SYSPUNCH,VOLUME=REF=*.SYSPUNCH,DISP=(CLD,PASS), X00008600
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=400) 00008700
//STAT DD DSN=*.INSTS,VOL=REF=*.INSTS,DISP=(CLD,PASS) 00008800
//TRANST DD DSN=ELTRANS,VOL=ELTRANS,UNIT=ELTRANS, 00008900
// DISP=ELDISP,SPACE=(TRK,1) 00009000

```



## NIPS 360 FFS

## PROCEDURES

```

//XOPSDEX PROC A=A,B=B,BSZFILE=,BSZFIL1=,BSZFIL2=, 00000100
// CL=',',CL1=',',CL2=',', 00000200
// CHKDSP=DELETE,CHKID='&&CHKID',CHKSP=0,CHKST=NEW, 00000300
// ISAM='DUMMY.FILE',ISAM1='DUMMY.FILE',ISAM2='DUMMY.FILE', 00000400
// JOBLIB='FFS.JOBLIB', 00000500
// LAB=SL, 00000600
// LIB='DUMMY.FILE',LIB1='DUMMY.FILE',LIB2='DUMMY.FILE', 00000700
// LIBDISP=SHR, 00000800
// NRMDSP=DELETE, 00000900
// PTFJOBL='PTF.JOBLIB', 00000910
// RGN=LOOK, 00001000
// SAM='DUMMY.FILE',SAM1='DUMMY.FILE',SAM2='DUMMY.FILE', 00001100
// STG=NIPW, 00001200
// TDISP=MOD,TRANS='&&TRANS', 00001300
// UCHK=NIPW, 00001400
// UISAM='(2314,P)', 00001500
// ULIB=2314,ULIB1=2314,ULIB2=2314, 00001600
// USAM='(TAPE9,,DEFER)', 00001700
// UTRANS=NIPW, 00001800
// VISAM=,VISAM1=,VISAM2=, 00001900
// VLIB=,VLIB1=,VLIB2=, 00002000
// VSAM=,VSAM1=,VSAM2=, 00002100
// VCHK=,VTRANS= 00002200
// * 00002300
// * CHARLES W. HICKISCH MAJOR,USA PRJ CODE=763NIPS BRANCH=431 00002400
// * DATE=MARCH 1, 1974 00002500
// * 00002600
//OP EXEC PGM=OP,REGION=&RGN 00002700
//STEPLIB DD DSN=&PTFJOBL,DISP=SHR 00002800
// DD DSN=&JOBLIB,DISP=SHR 00002810
//DATAFILE DD DSN=&ISAM,DISP=SHR,UNIT=&UISAM,VOLUME=&VISAM 00002900
//DATAFIL1 DD DSN=&ISAM1,DISP=SHR,UNIT=&UISAM,VOLUME=&VISAM1 00003000
//DATAFIL2 DD DSN=&ISAM2,DISP=SHR,UNIT=&UISAM,VOLUME=&VISAM2 00003100
//SAMFILE DD DSN=&SAM.S,DISP=SHR,UNIT=&USAM,VOLUME=&VSAM, X00003200
// LABEL=(,&LAB),DCB=BLKSIZE=&BSZFILE 00003300
//SAMFILE1 DD DSN=&SAM1.S,DISP=SHR,UNIT=&USAM,VOLUME=&VSAM1, X00003400
// LABEL=(,&LAB),DCB=BLKSIZE=&BSZFIL1 00003500
//SAMFILE2 DD DSN=&SAM2.S,DISP=SHR,UNIT=&USAM,VOLUME=&VSAM2, *00003600
// LABEL=(,&LAB),DCB=BLKSIZE=&BSZFIL2 00003700
//DLIB DD DSN=&LIB.L,DISP=SHR,UNIT=&ULIB,VOLUME=&VLIB,UNIT=&ULIB 00003800
//TLIB DD DSN=&LIB1.L,DISP=SHR,VOLUME=&VLIB1,UNIT=&ULIB1 00003900
//SOURCPR DD SYSOUT=(&A,&CL) 00004000
//PRINTER DD SYSOUT=(&A,&CL) 00004100
//SYSOUT DD SYSOUT=(&A,&CL) 00004200
//SYSPRINT DD SYSOUT=(&A,&CL),DCB=LRECL=133 00004300
//SYSUDUMP DD SYSOUT=(&A,&CL1) 00004400
//OPLINE DD SYSOUT=(&A,&CL2),DCB=LRECL=133 00004500
//OPIWCF DD UNIT=&STG,SPACE=(TRK,(2,1)),DCB=(BLKSIZE=400,LRECL=80, *00004600
// RECFM=FB),DSNAME=&CHKID.A,DISP=(,&NRMDSP,&CHKDSP) 00004700
//OPCOMREC DD UNIT=&STG,SPACE=(TRK,(1,1)), X00004800
// DISP=(,&NRMDSP,&CHKDSP),DSNAME=&CHKID.B 00004900
//OPLGOGO DD UNIT=&STG,SPACE=(TRK,(20,1)), X00005000
// DISP=(,&NRMDSP,&CHKDSP),DSNAME=&CHKID.C 00005100
//OPLEXEX DD UNIT=(&STG,SEP=OPLGOGO),SPACE=(TRK,(20,1)), X00005200
// DISP=(,&NRMDSP,&CHKDSP),DSNAME=&CHKID.D 00005300
//OPLEXDIR DD UNIT=(&STG,SEP=(CPLGOGO,OPLEXEX)),SPACE=(TRK,(20,1)), X00005400
// DISP=(,&NRMDSP,&CHKDSP),DSNAME=&CHKID.E 00005500

```

## NIPS 360 FFS

## PROCEDURES

```
//OPPUNCH DD SYSOUT=EB 00005600
//SLIB DD DSN=*.DLIB,DISP=SHR,UNIT=&ULIB,VOLUME=&VLIB 00005700
// DD DSN=*.TLIB,DISP=SHR,UNIT=&ULIB1,VOLUME=&VLIB1 00005800
// DD DSN=&LIB2.L,DISP=SHR,UNIT=&LLIB2,VOLUME=&VLIB2
// DD DSN=&PTFJOB.L,DISP=SHR
// DD DSN=&JOB.LIB,DISP=SHR
//CHECKDD DD DSN=&CHKID,DISP=(&CHKST,&NRMDSP,&CHKDSP),UNIT=&UCHK, 00006100
// VOLUME=&VCHK,SPACE=(CYL,(&CHKSP,1)) 00006200
//STAT DD UNIT=&STG,SPACE=(CYL,(2,1)) 00006300
//TRANST DD DSN=&TRANS,VOL=&VTRANS,UNIT=&UTRANS, 00006400
// DISP=&TDISP,SPACE=(TRK,1) 00006500
```

## NIPS 360 FFS

## PROCEDURES

```

//XSP      PROC  A=A,BLKSIZE=560,BSZFILE=,BSZNEWF=,      00000100
//          CL=' ',CL1=' ',DEN=,ISAM='DUMMY.FILE',      00000200
//          JOBLIB='FFS.JOBLIB',LAB=SL,LIB='DUMMY.FILE',  00000300
//          NBRBLK=200,RCN=100K,SAM='DUMMY.FILE',      00000400
//          PTFJOBL='PTF.JOBLIB',                      00000500
//          SAMOUT='DUMMY',SCRTSP=10,STG=NIPW,TRCH=,     00000600
//          UISAM=2314,ULIB=2314,USAM='(TAPE9,,DEFER)',  00000700
//          VISAM=,VLIB=,VSAM=,VSMOUT=,XDISP=OLD,        00000800
//          XINDEX='DUMMY.FILE',XUNIT=2314,XVOL=        00000900
//*****
//***      00001000
//***      ** 00001100
//***      THIS PROC IS USED TO GENERATE OR UPDATE A DISK RESIDENT ** 00001200
//***      INDEX DATA SET BASED ON THE DISK RESIDENT ISAM DATA FILE. ** 00001300
//***      ** 00001400
//***      //STEPNAME      EXEC  XSP,ISAM=AAAAAAA,VISAM='SER=XXXXXX', ** 00001500
//***                      XVOL='SER=YYYYYY',XDISP=(BBB),NBRBLK=DDDD ** 00001600
//***                      ** 00001700
//**      CHARLES W. HICKISCH MAJOR,USA      PROJ CODE=763NIPS BRANCH=431 00001800
//**      DATE=MARCH 1, 1974                00001900
//***      ** 00002000
//*****      00002100
//**      ** 00002200
//UTXSP      EXEC  PGM=UTNDXSPC,REGION=8RGN              00002300
//STEPLIB    DD  DISP=SHR,DSN=8PTFJOBL                   00002400
//           DD  DISP=SHR,DSN=8JOBLIB                     00002500
//INDEXPRT   DD  SYSOUT=(8A,8CL)                         00002600
//ISAMWORK   DD  UNIT=8STG,SPACE=(CYL,(10)),DCB=(DCSRG=IS,RECFM=VB, 00002700
//           LRECL=1000,BLKSIZE=1004,OPTCD=IYLR,CYLCFL=2) 00002800
//NEWFILE    DD  DSN=8ISAM,DISP=SHR,VOL=(8VISAM,UNIT=8UISAM 00002900
//SAMFILE    DD  DSN=8SAM.S,UNIT=8USAM,VOLUME=8VSAM,      00003000
//           DISP=(SHR,KEEP),LABEL=(,8LAB),DCB=(RECFM=VB,LRECL=1000, 00003100
//           BLKSIZE=8BSZFILE,TRTCH=8TRCH,DEN=8DEN)      00003200
//SLIB       DD  DSN=8LIB.L,VOLUME=8VLIB,UNIT=8ULIB,DISP=SHR 00003300
//           DD  DSN=8JOBLIB,DISP=SHR                     00003400
//SORTLIB    DD  DSN=SYS1.SORTLIB,DISP=SHR                00003500
//SORTWK01   DD  UNIT=8STG,SPACE=(CYL,(8SORTSP),,CONTIG) 00003600
//SORTWK02   DD  UNIT=8STG,SPACE=(CYL,(8SORTSP),,CONTIG) 00003700
//SORTWK03   DD  UNIT=8STG,SPACE=(CYL,(8SCRTSP),,CONTIG) 00003800
//SORTWK04   DD  UNIT=8STG,SPACE=(CYL,(8SCRTSP),,CONTIG) 00003900
//SORTWK05   DD  UNIT=8STG,SPACE=(CYL,(8SCRTSP),,CONTIG) 00004000
//SORTWK06   DD  UNIT=8STG,SPACE=(CYL,(8SORTSP),,CONTIG) 00004100
//SOURCPRT   DD  SYSOUT=(8A,8CL)                         00004200
//SYSOUT     DD  SYSOUT=(8A,8CL1)                        00004300
//SYSPRINT   DD  SYSOUT=(8A,8CL)                         00004400
//SYSUDUMP   DD  SYSOUT=(8A,8CL1)                        00004500
//UTSAMOUT   DD  8SAMOUT.DCB=(RECFM=VB,LRECL=1000,BLKSIZE=8BSZNEWF, 00004600
//           DEN=8DEN,TRTCH=8TRCH),DISP=(,KEEP),LABEL=(,8LAB), 00004700
//           UNIT=8USAM,VOLUME=(PRIVATE,RETAIN,8VSMOUT),DSN=8SAM.S 00004800
//XINDEX     DD  DSN=8XINDEX.X,DISP=(8XDISP,KEEP),UNIT=8XUNIT, 00004900
//           VOL=8XVOL,SPACE=(8BLKSIZE,8NBRBLK),          00005000
//           DCB=(BLKSIZE=8BLKSIZE,RECFM=F,KEYLEN=4,DSORG=DA) 00005100

```

## NIPS 360 FFS

## PROCEDURES

```

//XSTOIS PROC  A=A,BSZFILE=,BSZNEW=,CC=,CL=',' ,CL1=',' ,CYLOFL=1,DEN=, 00000100
//              INDEX=1,ISAM='DUMMY.FILE',JOBLIB='FFS.JOBLIB', 00000200
//              LAB=SL,NDISP=KEEP,OVFLW=5,PRIME=40,PGN=60K, 00000300
//              PTFJOBL='PTF.JOBLIB', 00000400
//              SAM='DUMMY.FILE',SEQNO=1,TRCH=,UISAM='(2314,P)', 00000500
//              USAM='(TAPE9,,DEFER)',VISAM='SER=CANCEL', 00000600
//              VOVFLOW='REF=*.DATAFILE',VPRIME='REF=*.DATAFILE', 00000700
//              VSAM= 00000800
//* 00000900
//* CHARLES W. HICKISCH MAJOR,USA PROJ CODE=763NIPS BRANCH=431 00001000
//* DATE=MARCH 1, 1974 00001100
//* 00001200
//STPI EXEC PGM=UTBLDISM,REGION=&RGN,PARM='&CC' 00001300
//STEPLIB DD DISP=SHR,DSN=&PTFJOBL 00001400
//          DD DISP=SHR,DSN=&JOBLIB 00001500
//DATAFILE DD DCB=(DSORG=IS,CYLOFL=&CYLOFL,BUFNO=5,BLKSIZE=&BSZNEW), 00001600
//          VOLUME=&VISAM, 00001700
//          DSNAM=&ISAM.(INDEX),SPACE=(CYL,&INDEX),UNIT=&UISAM, 00001800
//          DISP=(,ENDISP,DELETE) 00001900
//          DD DCB=*.DATAFILE,VOLUME=&VPRIME, 00002000
//          DSNAM=&ISAM.(PRIME),SPACE=(CYL,&PRIME),UNIT=&UISAM, 00002100
//          DISP=(,ENDISP,DELETE) 00002200
//          DD DCB=*.DATAFILE,VOLUME=&VOVFLOW, 00002300
//          DSNAM=&ISAM.(OVFLOW),SPACE=(CYL,&OVFLOW),UNIT=&UISAM, 00002400
//          DISP=(,ENDISP,DELETE) 00002500
//SAMFILE DD DSNAM=&SAM.S,DISP=SHR,UNIT=&USAM,VOLUME=&VSAM, 00002600
//          LABEL=(&SEQNO,&LAB),DCB=(RECFM=VB,LRECL=1000,BLKSIZE=&BSZFILE, 00002700
//          DEN=&DEN,TRCH=&TRCH,BUFNO=5) 00002800
//SYSPRINT DD SYSOUT=(&A,&CL) 00002900
//SYSUDUMP DD SYSOUT=(&A,&CL1) 00003000

```



## NIPS 360 FFS

## PROCEDURES

```

//XTABGEN      PROC  A=A, BLK=7294, CL=' ', CL1=' ', CL2=' ',
//              JOBLIB='FFS.JOBLIB', LIB=NONE, LIBDISP=CLD,
//              LIBSP='(2,1,5)', RGN=80K, SORTSP=8, STG=NIPW, ULIB=2314,
//              PTFJOBL='PTF.JOBLIB',
//              VLIB=
//              00000100
//              00000200
//              00000300
//              00000310
//              00000400
//              00000500
//* CHARLES W. HICKISCH MAJOR, USA PROJ CODE=763NIPS BRANCH=431
//* DATE=MARCH 1, 1974
//*
//TAB EXEC PGM=UTTABGEN, REGION=&RGN
//STEPLIB DD DISP=SHR, DSN=&PTFJOBL
//          DD DISP=SHR, DSN=&JOBLIB
//SYSOUT DD SYSOUT=(&A, &CL)
//SYSUDUMP DD SYSOUT=(&A, &CL1)
//SYSPRINT DD SYSOUT=(&A, &CL2)
//SORTLIB DD DISP=SHR, DSN=SYS1.SORTLIB
//SORTWK01 DD SPACE=(CYL, (&SORTSP), , CONTIG), UNIT=&STG
//SORTWK02 DD SPACE=(CYL, (&SORTSP), , CONTIG), UNIT=(&STG, SEP=SORTWK01),
//          SEP=SORTWK01
//SORTWK03 DD SPACE=(CYL, (&SORTSP), , CONTIG), UNIT=(&STG, SEP=SORTWK02),
//          SEP=SORTWK02
//SORTWK04 DD SPACE=(CYL, (&SORTSP), , CONTIG), UNIT=(&STG, SEP=(SORTWK01,
//          SORTWK03)), SEP=(SORTWK01, SORTWK03)
//SORTWK05 DD SPACE=(CYL, (&SORTSP), , CONTIG), UNIT=(&STG, SEP=(SORTWK02,
//          SORTWK04)), SEP=(SORTWK02, SORTWK04)
//SORTWK06 DD SPACE=(CYL, (&SORTSP), , CONTIG), UNIT=(&STG, SEP=(SORTWK01,
//          SORTWK03, SORTWK05)), SEP=(SORTWK01, SORTWK03, SORTWK05)
//SYSLMOD DD DSN=&LIB.L, VOLUME=&VLIB, UNIT=&ULIB, SPACE=(CYL, &LIBSP),
//          DISP=(&LIBDISP, KEEP), DCB=(RECFM=U, BLKSIZE=&BLK)
//          00000600
//          00000700
//          00000800
//          00000900
//          0001000
//          0001010
//          0001100
//          0001200
//          0001300
//          0001400
//          0001500
//          X00001600
//          0001700
//          X00001800
//          0001900
//          X00002000
//          0002100
//          X00002200
//          0002300
//          X00002400
//          0002500
//          X00002600
//          0002700

```



## NIPS 360 FFS

## PROCEDURES

```

//XTP   PROC   A=A,                                00000100
//       CL=' ',CL1=' ',CL2=U,CL3=C,CL4=S,CL5=T,      00000200
//       JOBLIB='FFS.JOBLIB',                        00000300
//       LIB='DUMMY.FILE',                          00000400
//       LIB1='DUMMY.FILE',                          00000500
//       PTFJOBL='PTF.JOBLIB',                      00000600
//       RGN=82K,                                    00000800
//       STG=NIPW,TPIMQ=TPIMQ,                      00000900
//       TPDUMP='SYSOUT=(A,,)',                    00001000
//       ULIB=2314,ULIB1=2314,                      00001100
//       UOMQ=2314,                                  00001200
//       VOMQ='REF=*.INMSGQ',                       00001300
//       VLIB=,VLIB1=,                              00001400
//       XINDEX='DUMMY.FILE',                      00001500
//       XUNIT=2314,XVOL=                          00001600
// *                                             00001700
// * CHARLES W. HICKISCH MAJCR,USA PROJ CODE=763NIPS BRANCH=431 00001800
// * DATE=MARCH 1, 1974                        00001900
// *                                             00002000
//TPMONSUP EXEC PGM=UTTPDRVR,REGION=&RGN           00002100
//STEPLIB DD DSN=&PTFJOBL,DISP=SHR                00002200
// DD DSN=&JOBLIB,DISP=SHR                        00002300
//SYSUT1 DD SPACE=(TRK,(0,5)),UNIT=&STG           00002400
//SYSUT2 DD SPACE=(TRK,(0,20)),UNIT=&STG           00002500
//SYSUT3 DD SPACE=(TRK,(0,5)),UNIT=(&STG,SEP=SYSUT1) 00002600
//SYSUT4 DD SPACE=(CYL,(0,1)),UNIT=&STG           00002700
//SLIB DD DSNNAME=&LIB.L,UNIT=&ULIB,VOLUME=&VLIB,DISP=SHR 00002800
// DD DSNNAME=&LIB1.L,UNIT=&ULIB1,VOLUME=&VLIB1,DISP=SHR 00002900
// DD DSNNAME=&JOBLIB,DISP=SHR                    00003000
//DATAFILE DD DISP=SHR,DSNAME=DUMMY.FILE,UNIT=(,P,DEFER) 00003100
//DATAFILE1 DD DSN=DUMMY.FILE67,UNIT=(NIPW,2),SPACE=(TRK,(0)),DISP=NEW 00003200
//DATAFILE2 DD DISP=SHR,DSN=DUMMY.FILE,UNIT=(2314,P,DEFER) 00003300
//DATAFILE3 DD DISP=SHR,DSN=DUMMY.FILE,UNIT=(2314,P,DEFER) 00003400
//SAMFILE DD DISP=SHR,DSN=DUMMY.FILES,UNIT=(2314,P,DEFER) 00003500
//EDCONSOL DD SPACE=(TRK,(5,,4)),UNIT=&STG         00003600
//EDITLIB DD DSN=DUMMY.FILE1,DISP=SHR             00003700
//INMSGQ DD DISP=SHR,DSNAME=&TPIMQ                 00003800
//OMSGQ DD DISP=SHR,DSNAME=&TPIMQ                  00003900
//OUTMSGQ DD SPACE=(TRK,0),VOLUME=&VOMQ,UNIT=&UCMQ   00004000
//AMSGQ DD UNIT=&STG,SPACE=(TRK,0)                 00004100
//SDCONSOL DD UNIT=&STG,SPACE=(2300,(10,,4))        00004200
//SDKNSET DD SPACE=(CYL,(1,1)),UNIT=&STG,DCB=(RECFM=F,BLKSIZE=1004) 00004300
//STATRECS DD SYSOUT=(&A,&CL)                     00004400
//SYSLMOD DD SPACE=(TRK,(20,,8)),UNIT=&STG,DCB=FFS.JOBLIB, 00004500
// LABEL=EXPDT=66366                             00004600
//SYSPRINT DD SYSOUT=(&A,&CL)                      00004700
//SYSABEND DD SYSOUT=(&A,&CL1)                    00004800
//TPDUMP DD SYSOUT=(&A,&CL1)                      00004900
//EDITDUMP DD SYSOUT=(&A,&CL1)                    00005000
//SYSONLIN DD &TPDUMP                             00005100
//XINDEX DD DSN=&XINDEX.X,UNIT=&XUNIT,VOL=&XVOL,DISP=SHR 00005200
//SHARDA50 DD DSN=NIPS.TPJCBQ,DISP=(MOD,KEEP)      00005300
//SYSIN DD DUMMY                                  00005400
//U DD SYSOUT=(&A,&CL2)                            00005500
//C DD SYSOUT=(&A,&CL3)                            00005600
//S DD SYSOUT=(&A,&CL4)                            00005700
//T DD SYSOUT=(&A,&CL5)                            00005800

```

NIPS 360 FFS

PROCEDURES

//SUBFILE DD UNIT=&STG,SPACE=(CYL,(0,1,10))  
//MENUSET DD DSN=DUMMY.FILEL,DISP=SHR

00005900  
00005910

## NIPS 360 FFS

## PROCEDURES

//XUTODE PROC	A=A,	00000100
//	CL=' ',	00000200
//	JOB LIB='FFS.JOB LIB',	00000300
//	PTF JOB LIB='PTF.JOB LIB',	00000400
//	LIB='DUMMY.FILE',	00000500
//	RGN=66K,	00000600
//	SDISP=SHR,	00000700
//	SNAP=,	00000800
//	SOURCL='DUMMY.FILE',	00000900
//	ULIB=2314,	00001000
//	USOURCL=2314,	00001100
//	VLIB=,	00001200
//	VSOURCL=	00001300
//XUTODE EXEC	PGM=UTODE,REGION=GRGN,PARM=&SNAP	00001400
//STEPLIB DD	DSN=&PTFJOB LIB,DISP=SHR	00001500
//	DD DISP=SHR,DSN=&JOB LIB	00001600
//SYSUDUMP DD	SYSOUT=(&A,&CL)	00001700
//SNAPSHOT DD	SYSOUT=(&A,&CL)	00001800
//SOURCPRT DD	SYSOUT=(&A,&CL)	00001900
//DDDPRT DD	SYSOUT=(&A,&CL)	00002000
//SYSPRINT DD	SYSOUT=(&A,&CL)	00002100
//SLIB DD	DISP=CLD,DSN=&LIB.L,UNIT=&ULIB,VOL=&VLIB	00002200
//SOURCLIB DD	DISP=&SDISP,DSN=&SOURCL.L,UNIT=&USOURCL,VOL=&VSOURCL	00002300

## NIPS 360 FFS

## PROCEDURES

```

//XTRDISK      PROC A=A,CL=',',CL1=',',JCBLIB='FFS.JOBLIB',      00000100
//              LAB=SL,RGN=60K,                                     00000200
//              PTFJOBL='PTF.JOBLIB',                             00000300
//              XFDISP=SHR,XFNAME=,XFUNIT=2314,XFVOL=,             00000400
//              ISAM='DUMMY.FILE',UISAM='(2314,P)',VISAM=,        X00000500
//              SAM='DUMMY.FILE',USAM='(TAPE9,,DEFER)',VSAM=,      X00000600
//              XTNAME=,XTUNIT='(TAPE9,,DEFER)',XTVCL=,           00000610
//              XTLAB=SL,STAT=NO,                                   00000700
//              BSZFILE=,TRCH=,DEN=                                00000800
//*****
//*              * 00000900
//*      THIS PROC IS USED TO TRANSFER A DISK-RESIDENT INDEX DATA SET * 00001000
//* TO TAPE. THIS OPERATION CONDENSES THE INDEX DATA SET. THE TAPE * 00001100
//* SO CREATED IS A SEQUENTIAL DATA SET CONSISTING OF VARIABLE LENGTH * 00001200
//* BLOCKED RECORDS THAT CONTAIN THE SOURCE DATA AND CONTROL * 00001300
//* INFORMATION FOR SUBSEQUENTLY RECONSTRUCTING THE SOURCE DATA. * 00001400
//*              * 00001500
//*      //STEPNAME EXEC XTRDISK,XFNAME=WW,XTNAME=XX,XFVCL=YY,XTVOL=ZZ * 00001600
//*              * 00001700
//*      CHARLES W. HICKISCH MAJOR,USA PRGJ CODE=763NIPS BRANCH=431 * 00001800
//*      DATE=MARCH 1, 1974 * 00001900
//*              * 00002000
//*              * 00002100
//*****              * 00002200
//*              * 00002300
//XTR      EXEC  PGM=UTNOXTR,PARM='DISK,&STAT',REGION=&RGN      00002400
//STEPLIB DD  DISP=SHR,DSN=&PTFJOBL      00002500
//              DD  DISP=SHR,DSN=&JOBLIB      00002600
//INDEXDAM DD  DSN=&XFNAME,UNIT=&XFUNIT,VOL=&XFVOL,DISP=&XFDISP      00002700
//INDEXPRT DD  SYSOUT=(&A,&CL)      00002800
//INDEXSAM DD  DSN=&XTNAME,UNIT=&XTUNIT,VOL=&XTVOL,DISP=(NEW,KEEP), X00002900
//              LABEL=(,&XTLAB)      00003000
//SYSPRINT DD  SYSOUT=(&A,&CL)      00003100
//SYSUDUMP DD  SYSOUT=(&A,&CL1)      00003200
//DATAFILE DD  DSN=&ISAM,      X00003300
//              UNIT=&UISAM,      X00003400
//              VOLUME=&VISAM,      X00003500
//              DISP=SHR      00003600
//SAMFILE DD  DSN=&SAM.S,      X00003700
//              UNIT=&USAM,      X00003800
//              VOLUME=&VSAM,      X00003900
//              DISP=(SHR,KEEP),      X00004000
//              LABEL=(,&LAB),      X00004100
//              DCB=(RECFM=VB,LRECL=1000,BLKSIZE=&BSZFILE,      X00004200
//              TRTCH=&TRCH,DEN=&DEN)      00004300

```



## NIPS 360 FFS

## PROCEDURES

```

//XRTAPE      PROC  A=A,CL=',',CL1=',',JOB LIB='FFS.JOB LIB',
//              XFLAB=SL,NBRBLK=50,RGN=60K,XFNAME=,          00000100
//              PTFJOBL='PTF.JOB LIB',                     00000200
//              XFUNIT='(TAPE9,,DEFER)',XFVOL=,XFDISP=CLD,   00000300
//              XTDISP='(NEW,KEEP)',XTNAME=,XTUNIT=2314,XTVCL= 00000400
//*****
//*              00000500
//*              00000600
//*          THIS PROC IS USED TO RECCNSTRUCT A DISK-RESIDENT INDEX DATA * 00000700
//* SET FROM A PREVIOUSLY UNLOADED SEQUENTIAL VERSION OF THE INDEX * 00000800
//* DATA SET. * 00000900
//* * 00001000
//* //STEPNAME EXEC XRTAPE,XFNAME=WW,XTNAME=XX,XFVCL=YY,XTVCL=ZZ * 00001100
//* * 00001200
//* CHARLES W. HICKISCH MAJOR,USA  PR CJ CCDE=763NIPS  BRANCH=431 * 00001300
//* DATE=MARCH 1,1974 * 00001400
//* * 00001500
//***** * 00001600
//***** * 00001700
//XTR          EXEC PGM=UTNDXTR,PARM=TAPE,REGION=GRGN      00001800
//STEPLIB DD   DISP=SHR,DSN=&PTFJOBL                      00001900
//          DD   DISP=SHR,DSN=&JOB LIB                     00002000
//INDEXPRT DD   SYSOUT=(&A,&CL)                          00002100
//INDEXSAM DD   DSN=&XFNAME,UNIT=&XFUNIT,VOL=&XFVOL,DISP=&XFDISP, X00002200
//          LABEL=(,&XFLAB)                               00002300
//SYSPRINT DD   SYSOUT=(&A,&CL)                          00002400
//SYSUDUMP DD   SYSOUT=(&A,&CL1)                          00002500
//XINDEX DD     DSN=&XTNAME,UNIT=&XTUNIT,VCL=&XTVCL,        *00002600
//          DISP=&XTDISP,SPACE=(560,&NBRBLK),             C00002700
//          DCB=(BLKSIZE=560,RECFM=F,KEYLEN=4,DSCRG=DA)   00002800

```